

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

**Sección Departamental de Arquitectura de Computadores y
Automática (Arquitectura y Tecnología de Computadores e
Ingeniería de Sistemas y Automática)**



**BEHAVIOUR BLENDING FOR MULTIPLE ROBOT
COORDINATED NAVIGATION THROUGH VIRTUAL
POTENTIAL FIELDS.**

**INTEGRACIÓN DE COMPORTAMIENTOS PARA LA
NAVEGACIÓN COORDINADA DE MÚLTIPLES ROBOTS
MEDIANTE POTENCIALES VIRTUALES**

**MEMORIA PARA OPTAR AL GRADO DE DOCTOR
PRESENTADA POR**

Santiago Cifuentes Costa

Bajo la dirección de los doctores

José María Girón Sierra
Juan Francisco Jiménez

Madrid, 2013



**UNIVERSIDAD COMPLUTENSE
MADRID**

FACULTY OF PHYSICAL SCIENCES

DEPARTMENT OF COMPUTER ARCHITECTURE AND AUTOMATICS

PHD THESIS:

**Behaviour Blending for
Multiple Robot Coordinated Navigation
through Virtual Potential Fields**

**Integración de Comportamientos para la
Navegación Coordinada de Múltiples Robots
mediante Potenciales Virtuales**

by Santiago Cifuentes Costa

Supervised by:
Professor Dr. Jose Maria Giron Sierra
Professor Dr. Juan Francisco Jimenez

Contents

1	Introduction	5
2	Previous Work	11
2.1	Mobile Robots	11
2.1.1	Potential Fields	12
2.1.2	Vector Field Histogram	16
2.2	Reactive and Deliberative paradigms	17
2.3	Behavior-Based Systems	18
2.3.1	Behaviour Based Control & Fuzzy Analysis	20
2.4	Multiple robots	22
2.4.1	Formations	23
2.5	Custom related works	27
3	General overview	29
3.1	Initial approach	29
3.2	Main Proposal	30
3.2.1	Action blending	30
3.2.2	Action description	32
3.2.3	World modelling	33
3.3	Test Cases	34
3.4	Results	37
4	Method Description	39
4.1	Mobile Robots	39
4.2	Algorithm Elements	40
4.2.1	Virtual Fields	40
4.2.2	Perception and data handling	43
	Descriptive Elements	43
	Situations	46
4.3	Decision process	49
4.3.1	Action Description	49
4.3.2	Employed Actions	52
4.3.3	Action overlapping	53
4.3.4	Action Processing	54

5	Studied Cases	59
5.1	Single Robot	59
5.1.1	System design	59
5.1.2	Simulation	63
5.2	Multiple, independent robots	67
5.2.1	System Design	67
5.2.2	Simulation	72
5.3	Unorganized Robots Group	79
5.3.1	System Design	79
5.3.2	Simulation	85
5.4	Robot formations	88
5.4.1	System Design	88
5.4.2	Simulation	101
	Formation Initialization	102
	Statistical Analysis	124
	Formation Manoeuvring	153
	Formation Switching	164
	Formation and Obstacles	169
	Virtual Structures	173
6	Real Experiments	177
6.1	Overview	177
6.2	Statistical Analysis	194
7	Conclusions	201
A	Simulation Environment	205
A.1	Main Elements	205
A.1.1	Simulation process	205
A.1.2	Information structures	206
A.2	User Interface	208
B	Experimental setup	211
B.1	Robot Hardware	212
B.1.1	Control Board	212
B.1.2	Radio Board	214
B.1.3	Driver Board	214
B.2	Robot Software	216
B.3	Robot positioning	216
B.3.1	Local Positioning	217
B.3.2	Visual Positioning	217
B.3.3	Merging	221
C	Custom related works	223
	Bibliography	225
	List of Figures	235
	List of Tables	237

Chapter 1

Introduction

Robots are amazing machines: capable of nearly everything from tending our house, helping our elders, do the harsh labours for us, and even save the world! –or some times conquest it– ... or at least that is the idea that we all have from the media, the science fiction and other futuristic fantasy.

The truth is that robots are amazing machines, capable of lots of things, but they are still being developed, and need a lot of refining before they can do all those marvellous things.

In the last decades we have experienced a quick growth on the electronic machines that help us in our daily live and our industry. Radio, television, fridges, vacuum cleaners, computers, music players, mobile phones ... we have grown accustomed to technology.

At each step, we expect technology which depends less in our expertise and becomes more capable of adjusting to our needs with the minimal knowledge on our side.

We expect more independence and autonomy from our machines, partially because detailing all small aspects and solving all small conflicts is tiresome, requires expertise, and will degrade the experience of using that technology.

But that need of expertise and the need to solve the small conflicts is partially due to the way we thought and, therefore, the way we use to express our desires and ideas when we are presented with a complex enough task.

When we thought how to solve a complex task, it is not usual to find someone who immediately considers all aspects and all interactions and everything involved with that task. The common approach will be to solve isolated aspects, one at a time, and then refine the partial solutions looking for conflicts among them. The expertise is needed to solve those conflicts and to be able to see that no aspect have been left behind.

Therefore, we look for machines which are able to work with our initial, unrefined, isolated considerations, where the machine is the one that provides the expertise, solve the conflicts and fill the gaps.

Probably the idealization of these machines are the robots. And probably we will reach a point where the robots were to computers as our smart phones are to Marconi's radio.

But for this, we still have to solve some problems. One of them being to improve the ability of a machine to take a collection of independent statements –with their small contradictions– and put them together to perform the task that we expect –and not just the one we have stated–.

Topic overview Robots are just a smart combination of mechanics, electronics, and logic, but it is the level of that logic which separates common machines from what we consider robots. Without a high enough logic, we just have some kind of automated machine, such as a washer or vacuum cleaner.

Some aspects of robotics are mainly solved and easily available:

- Electric motors and mechanical structures have highly evolved through its use in multiple industries, and are now present in many devices surrounding us. So, robot bodies can be built without many problems –the main problem here, is how to power them for long enough–.
- Computational power to sustain the logic part of the robot is evolving in a daily basis, smaller and more capable computers are constantly being released –for some tasks, parallel computation stills need to be improved, but the ability to perform quick computations is there–.
- Sensors are the way through which a robot is able to sense the worlds around it, in the same way our senses provides us such information. As the electronics and computers become ubiquitous, a great variety of sensors have appeared and become affordable. While some electronic sensors needed to mimic our own sensing capabilities are still being improved –taste, smell or touch– other sensors such cameras and microphones have reached the resolution level used by our own senses.

The main lack of the robotics field is, nowadays, at the logic side of the robot. How the information of the sensors must be analysed, how the mechanical elements must be coordinated and how to provide the intelligence that we assume in a robot to be a robot and not just an automated machine.

An element of this missing logic capability is, in some aspects, the ability to put to work together all the elements found in the robot. The problem is not usually found in the lack of computational power, but in the method –or the lack of a good method– used to instruct the robot on the behaviours that must display.

What can be usually found in current robots is that each problem is confronted in a specific way, and when a robot must face many problems at once, the different specific ways studied to solve each of the problems are not compatible, or do need a complete rework in order to have them acting together.

General proposal A general solution which was valid for all the aspects in robotics is not easily achievable because of the broadness of the robots applications. However, the exploration of possible solutions for some specific applications may give us a chance to find a principle which can be expanded and applied to other areas of robotics.

Along this Thesis a way to evaluate the robot situation and provide the different actions that it should take in order to fulfil its task is proposed.

The work is focused in the topic of robot navigation, a fundamental area that is present in most robots.

While the main proposal is a way to coordinate the different elements that should be considered by an individual robot to navigate accordingly with its imposed task, the applications of the proposed method shown along the Thesis ranges from the single robot navigation to the coordination of multiple robots moving together.

Thesis structure The text of this Thesis begins, at this first chapter, with some overview and general motivations of the topics related with the work presented along this document.

To this introductory chapter, it follows a review of the main authors and topics that have influenced those fields of robotics considered along this work. This review of literature is accompanied with small notes on how different aspects on the development of robotics have influenced and are related to this Thesis.

After that, in the third chapter can be found a personal reflection of the previous literature and the subsequent thoughts on how the field of robotics can be enriched. It follows a quick overview of the proposed algorithm, their main elements, the test and their results in a condensed form, without any formalism, but that will help to see the proposed idea and work in a easy way.

It is then, at the fourth chapter, where the proposal of this Thesis is done. With an approach that intends to be wide enough to be valid for many robots, the different elements taking part on the proposed method are explained. Once the different elements are known, the main proposal is done, and is followed by a general –but focused in the navigation– method to apply it.

Following the Thesis proposal, the fifth chapter illustrates how to work with the proposed method. For this, four different scenarios are presented and the method applied to them. The increasing complexity of the presented scenarios goes from the navigation of a simple robot to the coordinated navigation of a group of robots following a given formation scheme. This increasing complexity is used to illustrate the different aspects of robot navigation and how they can be handled by the proposed method. It also shows how the proposed methods deals with reusability and the increases in complexity without the need to completely rewrite the robot control structure.

While the scenarios solved in the fifth chapter are evaluated through simulation, the applicability of the method for real world robots is shown in the sixth chapter. There, the last and most complex of the simulated scenarios, the robots navigating in formation, is tested again using real robots. Being the results of the real robots test similar to those obtained through simulation, the results obtained through simulation are validated, supporting the ideas resulting from the analysis of the simulation results.

At the view of the obtained results, some conclusions are reached at the seventh chapter. Also some reflections are done on the applicability of the proposed hypotheses to other areas and the future work that can be done following this research line.

Finally, the Appendices provides some detail on the tools used for simulation, the real robots built to validate the presented work and how the testing have been done.

Contribution The work done along this Thesis rest in the many developments found in previous studies. It takes ideas from many authors, inherited which were considered the better points of view, and the ones that better fit with my own line of reasoning.

Most of the individual aspects of this Thesis are not original work, what can be considered original is the combination of all them into a single methodology, and how they are applied together, following the same principles, to solve a wide range of problems.

Also, the problems studied to test the proposed method are not all problems already solved. The way in which they are handled through this Thesis provides a solid solution for them, and with generality enough to be employed in conditions not exactly equal, and still expect similar results. This can be expected because the results obtained from the tested problems support some ideas that, while intuitive, are not easy to demonstrate in a formal way.

Thanks This Thesis would not been possible without the support of many people.

My parents, who let me fly in the wings of imagination with all those books, mainly fantasy, and let me spend hour after hour with all kind of constructions sets, but who also pull me to the ground to study and to seek how knowledge can expand my games.

My friends, with whom I have had the most bizarre and crazy discussions which amazingly have led to notable conclusions, and whose have followed and shown full interest with my ideas and crazy robots, even when they do not understand them at all.

The people of the Computer Architecture and Automatics department at the Physics Faculty on the Madrid Complutense University, where all my work have been done and from which people I've learned a lot.

The people of the SmartFuel project, because while they have not been directly involved in this Thesis, the collaboration with them have provided me a insight from the industry side which cannot be easily acquired while working only at the university.

All the people who have been in the laboratory 237 along all these years. Some of them have been able to end their projects, some not, but all have been good companions along the trip, and the discussions that have taken place there after lunch have solved -many times- our works and the world.

My two counsellors, Dr. Jose Maria Giron-Sierra and Dr. Juan Francisco Jimenez, which have followed all the steps of this Thesis and with whom I have evolved from a robotics enthusiast with more ideas than knowledge to a robotic enthusiast with more ideas than time.

And I have to specially thank to my wife. Who support me and the beginning of the process, when I was not sure of what to do, or if it worth it. Who have bore with my moments of focused isolation, and have withstand my the urge to discus with someone –her– the ideas running in my head.

To her I have to thank, because she have let and encourage me to keep playing with robots.

Chapter 2

Previous Work

The work of many people have influenced on this work, and some of them have been critical in the development of the different aspects of this thesis. However this work do not follow strictly any specific approach, but it is built out of many interesting aspects found in different works. Along the following review of the previous works related with this Thesis, only the main contributions and key point ideas have been referred, since the derived literature is quite extensive.

My first memories about robots –and the initial love for them– were the adventures of *Norby The Robot*, a small collection of books for children written by Janet and Isaac Asimov. I do not know who was Isaac Asimov at that time, actually I was not aware that those books were written by Asimov after many years later, when I discovered the books again in a shelf. Many science fiction writers but especially Asimov deserves his mention among those who have inspired this work, maybe it would not be such work without them.

2.1 Mobile Robots

Along history, many automata and mechanical devices have been built, but it was not until the beginning of the XXth century that the first machines that can be called robots were built. Those initial automata work on predefined sets of movements or actions, but it was not until those actions were based on the environment sensed by the machine, so the machine can act based on its perceptions, that we can call those machines *Robots*.

Among the first examples of machines that were able to react to its environment we can find the *Electric Dog* built by John Hammond, Jr. and Benjamin Miessner near 1912. The *Electric Dog* shows a phototropic behaviour –it moves towards the light– as many of the first robots built by today enthusiasts –as it was mine–.

In the late 1940s the conjunction of control theory, information science and biology adds up to the development of *cybernetics* by Norbert Wiener and others, with the vision of an artificial organism ruled by mathematics and control laws able to express natural behaviours.

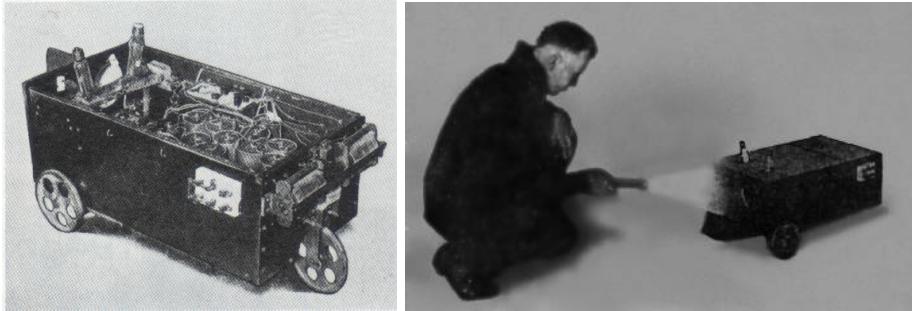


Figure 2.1: The electric dog and a test showing how it follows the light

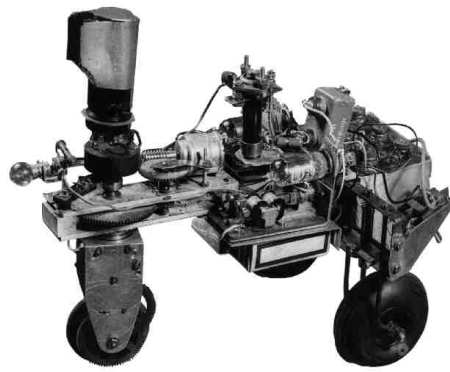


Figure 2.2: Elise, Walter's second tortoise. One of the first robots with multiple behaviours

The first machine that can be considered a robot with multiple behaviours able to autonomously move around, acting based both in its environment and its internal perceptions, was designed and termed *Machina Speculatrix* by W. Grey Walter following Wiener's principles. Several mechanical implementation were done around 1948, and they are known as the Grey Walter's Tortoises. The several behaviours that this robot exhibits were: Seeking the light, Head toward weak light, Back away from bright light, Avoid obstacle and Recharge battery; the different behaviours were triggered by different internal and external conditions and the result was a 'natural' behaviour in which the tortoise moves safely around the room and recharges itself when needed.

2.1.1 Potential Fields

Potential fields and virtual forces were initially introduced for its use in industrial robotic arms and other fixed robotic manipulators. In that context the objective was to move the robot effector to a desired position, where the main constraints were imposed by the surroundings and by the robot body itself. Conforming to the articulation limits and avoiding the actuator to go through the space occupied by the robot body where the main points. Also avoiding the

static obstacles in the robot working area, especially the object in which the robot was working, was considered.

One of the main driving forces of the potential field was O. Khatib who resumes the method with *"The philosophy of our approach can be schematically described as follows : The manipulator moves in a field of forces. The final position to be reached is an attractive pole for the terminal device, and the obstacles are repulsive surfaces for all the manipulator parts"* in [Khatib78]. Its continuous work on this approach, like [Khatib80] and the extrapolation of its use for mobile robots [Khatib85], [Khatib86] makes him to figure among the fathers of the potential field method, however he refers in [Khatib78] to M. Renaud [Renaud76] as the source of introducing an artificial potential in the equations that rules the robot manipulator dynamics.

The origin of the potential field method being called so, is that the formulation used to analyse the robot motion was through the use of Lagrange equations.

A robotic arm, with a kinetic energy T evolves in a potential U according with Lagrange equations. This potential U was usually restricted to the gravity potential; with the addition of an artificial potential –that considers the repulsion of the robot body parts to the obstacles in the environment– results a minimal energy path between the robot effector position and its goal.

This new path was no longer the straight line between those two points as can be seen in figure 2.3. The approach without the artificial field was not necessarily a straight line, since it considers also the inertia of each segment of the robot arm, but that initial approach does not provide any restriction about going through the solid bodies present around the robot arm.

The origin of the virtual forces are therefore a consequence of this Lagrangian formulation. The generalized forces obtained, once the artificial potential field is introduced, includes then a new contribution due to this artificial potential.

The expression for the artificial potential has suffered several modifications along time. The initial expression used by Khatib, in eq. 2.1, for the attractor leads to a resulting force equivalent of that of a spring. For the repulsor Khatib uses a similar expression but with the inverse distance, as shown at eq. 2.2, where ρ_0 represents a limit distance so the potential and forces were zero beyond that distance; it is usual to find $\rho_0 = \infty$ so the field expression is continuous and unique along all the domain

$$U_{x_d}(x) = \frac{1}{2}k(x - x_d)^2 \Rightarrow F_{x_d} = -k(x - x_d) \quad (2.1)$$

$$U_o(\rho) = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho} - \frac{1}{\rho_0})^2 \\ 0 \end{cases} \Rightarrow F_o(\rho) = \begin{cases} \eta(\frac{1}{\rho} - \frac{1}{\rho_0})\frac{1}{\rho^2} & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases} \quad (2.2)$$

However these expressions, once applied to mobile robots show one of the main problems of the artificial potential method: the local minima traps. Not long after the publication of the method some authors, [Krogh84], [Lyons86], [Borenstein89], [Koren91], detected some limitations of the artificial potential fields, being the most critical ones the appearance of local minima in the field and the occurrence of cyclic oscillations. For the local minima, on certain alignments between the goal and the obstacles, the field gradient points towards a local

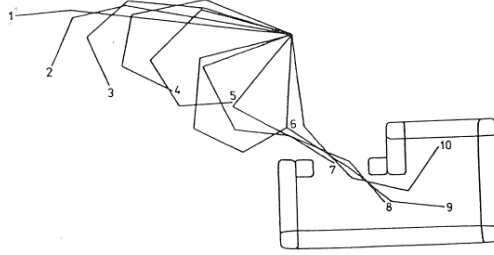


Figure 2.3: Robot arm movement result exposed in the work of Kathib

minima instead of the goal, so the robot is trapped there and unable to reach the goal. The occurrence of cyclic oscillations was mainly found in narrow passages where they can lead the robot to collide with the walls.

To avoid the local minima problem, Krogh in [Krogh84] proposes a generalized field expression that considers the position of the goal, the obstacles and the robot position within the map. Using these elements the field can be tuned so the only minimum was located at the goal position. The main difficulty of this proposal was that the artificial field do not remains static and need to be tuned at each step. Another example on the effort to build an artificial field free of local minima was [Lyons86], who takes apart the attractive part and the repulsive one. In this approach there are still local minima in the repulsive field, but they do not affect in the same manner the behaviour of the manipulator since they handle separately the actions due to the obstacles and the ones related to the goals.

Later modifications to the artificial potential expression include the consideration of the robot speed when Krogh in [Krogh86] or Borenstein and Koren in [Borenstein89] uses the method in real fast robots. There, the inclusion of the robot speed boosts the relevance of the obstacles in the trajectory of the robot, so the avoiding movement begins earlier and the robot was kept far from the local minima regions.

Other approaches to avoid this local minima and oscillation problems included analytical methods where the trajectory of the robot is reviewed to see if it is oscillating near a position different to the goal; algorithmic solutions where the robot is given some artificial inertia to make it able to navigate through some of this local minima; or methods where the obstacles also produces a rotational force around them so the pushes the robot to circumnavigate them [Arkin90], [Slack91], [Gat93]. More sophisticated approaches can be also found like [Siemiatkowska94], where the artificial potential is built after fluid diffusion equations, the obstacles potential built as superquadrics by [Khosla98] or the virtual obstacles placed in the map to shadow the local minima affecting the robot used by [Park03]

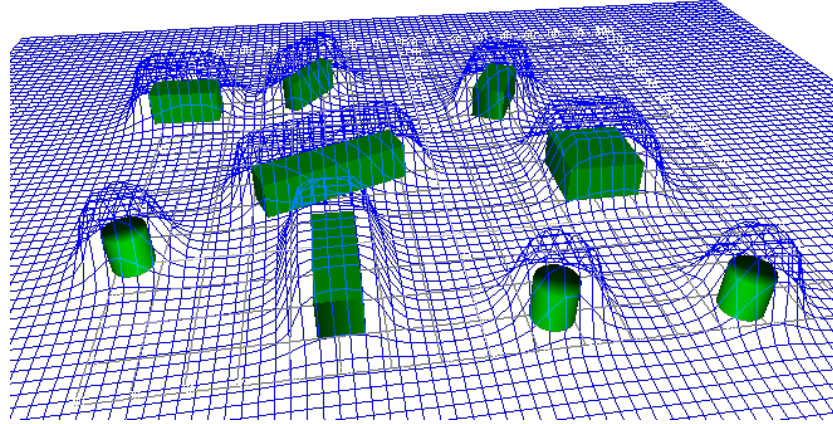


Figure 2.4: Potential field on a scenario with several object, its representation is like covering everything with a soft sheet

While potential fields suffer from known problems when used as the only method for the navigation of a robot, the intuitiveness of its representation and evolution when used in reduced and controlled situations have resulted in a extensive use of this method as a support technique to complement more complex methods.

Path relaxation methods, like the ones proposed in [Krogh86], [Thrope84] or [Quinlan93], uses a potential field method as a complement to smooth the results of graph-based planning methods. In graph based methods some characteristic points or lines are selected along the space and some optimal finding method is applied to link the current robot position and goal through the use of these characteristic elements. The results, while optimal in their metric and quick to obtain, are often a collection of straight lines that are not smooth enough for a real robot to follow. The potential field method is used afterwards to soften these trajectories so they are more achievable for a real robot.

It is also usual to find approaches where a higher level algorithm takes care of the global navigation analysis and a potential field method is left to deal with the immediate surroundings or to deal with sub problems not considered in the general path plan. In works like [Krogh86] an ideal robot and a real robot are considered; given a path to be followed by the ideal robot, a potential field is used to link the ideal robot position over the path with the real robot position on the world, so the real world robot can adapt to unforeseen elements, motion inertia, etc. As for sub tasks, [Balch00], [Reif99] or [Kang11], use the potential fields in order to keep a group of robots together while the group as a whole follows an externally given path. Multirobot manipulation based on potential fields is considered by [Song02] and multirobot coordination through potential fields can be found in [Vail02].

In spite of their problems, potential fields have been widely used from the very beginnings of applied robotics to nowadays due to its simplicity, intuitiveness and results in controlled spaces. This work, while not using the potential

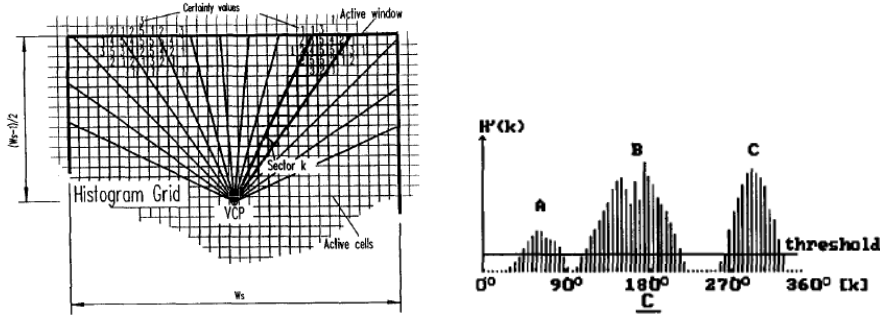


Figure 2.5: Polar map built out of the Certainty Grid representation and flat result from the work of Borenstein

fields directly to navigate, do use intensively the potential field abstraction as a mechanism to contain the information of the various elements around the robot in order to be able to navigate. The intuitiveness of the potential field representation allow us to handle undefined amounts of elements that can influence the robot navigation in a simple and elegant way.

2.1.2 Vector Field Histogram

The *Vector Field Histogram* was introduced by Borenstein and Koren after their (unsatisfying) experimental trials using pure virtual field methods with fast moving robots in [Borenstein89].

In order to build the vector field histogram out of experimental data, they first use the *Certainty Grid* introduced by Moravec and Elfes in [Moravec85]. The Certainty Grid is a static Cartesian grid placed over the robot surroundings where each grid element holds the probability, obtained through sensor measurements, of finding an obstacle on that position. The final shape obtained through the Certainty Grid was quite similar to the result of building the artificial potential at each grid element, but using experimental data instead of the maps used by the previous potential field methods.

Instead of using the Certainty Grid for building a potential field as in their previous experiments, in [Borenstein90] they built a polar map representing the instant probability of collision in each direction surrounding the robot.

In the final phase of the navigation algorithm, [Borenstein90] analysed the built vector field histogram and chose a direction leading to a region free of obstacles and nearer to the current robot heading direction and goal.

This idea is later revised by Ulrich and Borenstein in [Ulrich98] where the method speed is improved through threshold binarization of the vector field and better final analysis. One further step can be found in [Ulrich00], where a short term motion prediction is introduced and the result is improved by taking the direction leading to the best immediate path. Other works have followed the idea of the vector field histogram such as [KhatibM96] or [Minguez04].

The current work takes the same idea of building a instantaneous image of the robot state for each direction, using the data extracted from of a more

permanent map. Several aspects considered by the robots are hold this way prior to its processing to establish the final robot movement. However, the employed data used in this work are not a direct representation of the robots surroundings.

2.2 Reactive and Deliberative paradigms

One of the main differences between the Vector Field Histogram and the Potential Field methods, aside from the obvious ones, is the level of analysis and reconstruction of the available information and how the algorithm works not directly from sensor data but from a representation of the world built out of the sensor data –one level of abstraction is introduced on the vector field histogram data–.

The less analytic methods, those which work directly with sensor data, are called reflexive methods in [Arkin98], and were strongly defended by Brooks, [Brooks86]. In the reflexive methods the output of the system is plainly linked to the sensors input with none or little analysis of the global situation or previous history: A hammer hits the knee, the knee moves up. In robotics this can be found in early, very early robots and automata as the *Electric Dog* or Walter's tortoises [Walter53]. Nowadays a current called BEAM robotics, originated by Tilden [Tilden95] works on the basis of behaviour originated out of specific configurations of a "nervous system" which is in most cases a pure analogue circuit.

As the analysis of the data is increased the computational cost arises, but the sensor data can be merged in some world model which eases the coexistence of many sensor readings and provides some frame of reference for the construction of more complex robot algorithms. Higher level of abstraction on the world representation is associated with deeper analysis of that world in order to establish some desired behaviour in the robot. This also slows the reaction times of the robot on sensory data but allows predictions on how the situations will evolve (as long as the world do not change), thus providing the tool to obtain optimal solutions for the problems faced by the robot.

While the pure reactive, or reflexive, methods react instantly on sensory data but do not plan even a cycle ahead, pure deliberative methods will take data once, built a world model and perform all the task based on that unique glance of the world. Pure deliberative approaches work only in worlds that do not change along time, like static factory lines, but they are inappropriate in mobile robotics, were the world frequently changes. There are two common approaches to combine reactive and deliberative methods. The first is to develop a single method half deliberative half reactive which plans moderately ahead but also update its world model frequently enough to be able to react. The second approach is to run simultaneously two methods, a long term deliberative planner and an immediate distance reactive navigation scheme, where the reactive navigation scheme takes its goals from the long term planner but also supervises the viability of the established plan so when the robot, due to the world changes, moves too away from the initial plan, the reactive method asks for a new plan to the deliberative method. The work of Ulrich and Borenstein in [Ulrich00] can

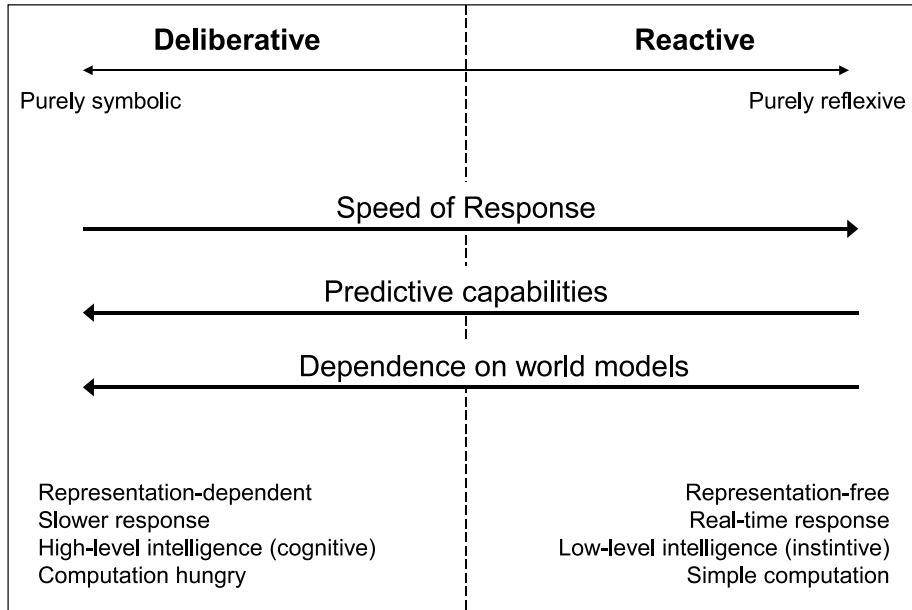


Figure 2.6: Deliverative Vs Reactive Approaches, from L. E. Parker courses

be taken as an example of the first approach, while the work of Quinlan and Khatib in [Quinlan93] will fall under the second approach. The second approach is more frequent in literature because it allows the use of well known methods for each specific task, while only requiring the effort to develop a way to identify the points where a new plan is needed.

The current work is pointed towards the second approach and focused in the immediate distance navigation scheme. A local navigation method is developed along this text. It is assumed that the goals are provided by some external means, being a human operator or a long term planner, which is not discussed along the work.

2.3 Behavior-Based Systems

The initial literature on robot navigation just deal with going from point A to point B avoiding collisions with obstacles. These scenarios were inherited from the research about robot manipulators, which were widely studied at the time due to their increasing industrial use. As autonomous robots becomes a study field by itself, it was also influenced by other research areas. Biology, Psychology and Brain Theory were regarded as source of information about how to develop robots able to handle more complex situations and different kinds of information. Looking to these other research fields has also another motivation, or maybe consequence: how to express the robot operation in a more "natural" or "human" way as opposite to the pure mathematical approach used in industrial robot manipulators, which model their world and behaviour directly

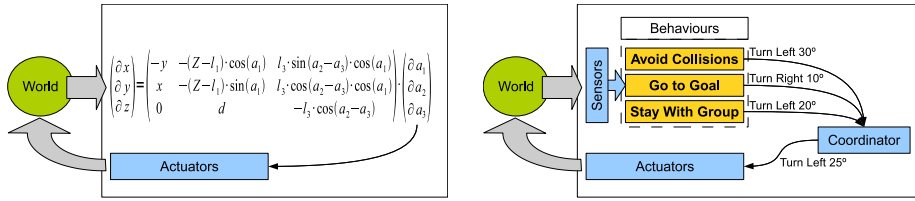


Figure 2.7: Classic approach –using robot dynamic equations– and Behavioural approach –evaluating objectives independently–.

through kinematic equations. A good review about the different implications of behaviour based control can be found in [Mataric92b]

Robot manipulators have the advantage of dealing with very deterministic environments and very little uncertainties while mobile robots often deals with just the opposite: unknown environments prone to changes and high amount of uncertainties, therefore the kinematic approach of robot manipulators was not suitable for mobile robots. A Biology/Psychology inspired approach, later called *Behavior-Based* robotics, was developed where the "actions" of the robot were inspired from some kind of "motivation" or "intention". Among the seminal works of this approach were Brooks [Brooks86] and Arkin [Arkin89] who introduced the two main different approaches followed later.

Both, Arkin and Brooks, developed an approach where the robot evaluates, independently, several aspects of the scenario in order to maintain specific behaviours, e.g. "avoid collisions" or "keep in the path". In this way, all the elements to be considered by the robot to perform its task were evaluated individually, and each of them proposes a command to apply to the robot motors/actuators in order to remain within the behaviour parameters or when the situation evaluation was not satisfactory. An illustration about the behavioural approach and the classical approach can be found in figure 2.7

The main difference between Arkin and Brooks was how the commands, resulting from evaluating the different situations, were put together in order to provide a single instruction to the robot motors/actuators. Brooks approach uses a hierarchical priority approach called *subsumption* architecture. There, the commands resulting from lower priority situations, are evaluated by higher priority situations which can block those commands that conflict with their own outputs. Meanwhile Arkin, influenced by the potential field methods, performs a weighted sum of the commands proposed by the different behaviours, so all the behaviours contribute to the final solution.

Both methods have been the reference for later works, but also both methods have a common difficulty: they need a supervision mechanism to establish the priority –for Brooks– or the weight –for Arkin– because the two of them provides poor results when fixed priorities/weights are used. This supervision mechanism must hold the knowledge of the overall task of the robot to provide proper results, thus, the knowledge of the robot task is split between the behaviours definition and the supervision function, so those two definitions must be tuned accordingly.

The present approach could be also categorized as a Behaviour-Based system since it evaluates a collection of situations –specific for each case of study– and each situation define a set of actions for the robot to take. But it differs from these seminal works in that the independent behaviours do not propose commands applicable directly to the robot actuators, but they propose *actions* which are mapped to an action space, bigger than the actuator space. These extra dimensions of the action space are built in a way that provides a discrimination factor to those behaviour reactions that are conflicting in the actuator space.

The supervision function used in the classic Behavior-Based robotics is here replaced by a function used to map the action space to the actuator space, with the characteristic that this function is fixed and do not depends on the specific robot overall objective as long as it can be expressed through the same action space. The knowledge usually located at the supervision function is now stated in the behaviour description though the use of the extra dimensions on the action space. In this way all the system knowledge is located at the behaviour description.

2.3.1 Behaviour Based Control & Fuzzy Analysis

Classical Behaviour-Based systems do take punctual information from the sensors, each behaviour evaluates this information and outputs a single action to be taken. But in [Rosenblatt89] is stated that by using this punctual output for each behaviour there was a loss of information about the internal knowledge of each behaviour, thus difficulting the work of the supervision method, aimed to provide a single output to the robot actuators out of the many outputs of the individual behaviours.

Rosenblatt original proposal –which follows the *subsumption* architecture from Brooks– was to increase the number of behaviours by providing the evaluation of slightly different variations of each one of the original behaviours. This proposal drives to the evaluation of a group of outputs –one output from each variation– and their combination to obtain a single result. However, the obtained result was comparable to the results of a multivalued logic analysis operated on the original behaviour. This leads to the apparition of behaviour-based methods based on fuzzy logic approaches. Multivalued logic allows the behaviours to state their desired actions in a non-punctual way as depicted in figure 2.8. Through this kind of action expression it was easier to combine the different behaviour outputs in a way that satisfy most behaviours at once. Many different approaches following this idea were tried like the ones in [Saffiotti95] or [Michaud97]; a good review can be found in [Saffiotti97].

The use of fuzzy logic for behaviour based robots also leads to better formalization for the situation awareness of the robots (here situation awareness is conceived as the method to evaluate the situation or situations affecting the robot) and which behaviours must be considered for the robot reaction. By using fuzzy logic, a relevance level is assigned to each behaviour instead of the fully active/inactive used in previous approaches.

The works that combine behaviour based robots and fuzzy methods were able to provide better results than comparable non-fuzzy methods, but one problem remains: how to combine the outputs of conflicting behaviours. To cope with this problem, the initial approaches from Brooks and Arkin remains still

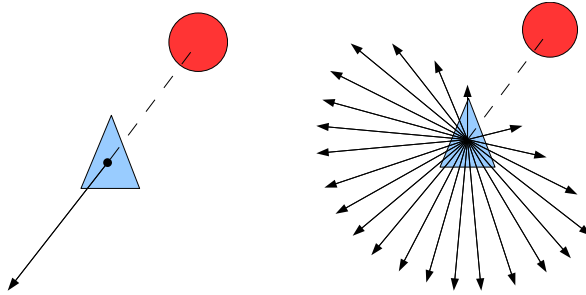


Figure 2.8: Uni-Valued and Multi-Valued output from an *Avoid Collision* behaviour

valid, referred as *arbitration* and *command fusion* in [Saffiotti97]; [Prijanian99] contains a deep review on the different techniques of behaviour coordination mechanisms.

From the many works that combines fuzzy analysis and behaviour based control, it is pertinent to mention the work of Riekki & Rning in [Riekki96] for its similitude with some aspects of this Thesis. Riekki & Rning, following the main idea found in [Rosenblatt89], states that there is a loss of information from the behaviour evaluation to the behaviour action specification, and this loss makes difficult the later action fusion.

In their work, Riekki & Rning, as in many other fuzzy approaches, states the actions intended by each behaviour as a action map, a map containing the directions around the robot with a level of *desirability* of the action for each direction. But what characterises their proposal was that they built two maps for each action, one for the positive action, i.e. *goto* some direction, and one for the opposite action, i.e. *dont-goto* some direction. Each behaviour can fill one of these maps and the final command was composed by a weighted sum of those maps. In this way, an easy method to devalue an specific direction was introduced for multivalued analysis, in the same way as the vector sum method approach in Arkin allows to avoid the direction of an obstacle by proposing to move in the opposite direction.

The current Thesis also uses this multivalued approach, both for the situation awareness and for the action declaration of the different behaviours through action maps. It is also proposed, with a similar approach as in [Riekki96], the use of several action maps as behaviour outputs related for each single robot action/actuator in order to increase the amount of information handled by the command fusion process. The main difference with Riekki & Rning is that the current approach considers those extra maps as *motivation* maps for the actions. In this way the outputs of behaviours that are typically considered conflicting do not share the same map, but, since they have different *motivation* they do state their actions in different action maps. Also the maps that contains opposite motivations, i.e. *goto* & *dont-goto*, are not handled by a weighted sum but more like in a hierarchical or veto approach.

2.4 Multiple robots

As the study of single robots was acquiring a good ground, the interest for multiple mobile robot systems arose. The motivation for the multi robot systems is the same concerning the human crews or work groups: some tasks are easier, or even only possible, to be fulfilled by the simultaneous work of several individuals. The efforts on multiple mobile robot systems are focussed on three main topics:

- **Reconfiguration:** The capability to adapt and combine the different robots skills to become able to perform a task.
- **Coordination:** The way for multiple robots to work together at the same time without interfering among themselves and, when possible, optimizing the results. This includes the wide topic of motion planning and multiple vehicle navigation, which is considered along this Thesis.
- **Cooperation:** The explicit distribution of the sub-tasks among the robots to fulfil a global task which cannot be completed by any individual. Where the distribution is done by the same robots.

Other ways to categorize the works in multiple robot systems can be established depending on their origin or the specific tasks that are achieved. In the reviews of [Parker00] or [Arai02] different categories can be found along with a re-compilation of the main works on those areas, and an extensive taxonomy based on the different robot capabilities can be found in [Dudek96].

While the most demanding part in this Thesis lays in the coordination of the multiple behaviours working inside a single robot, this effort is oriented to the coordination of several robots sharing the same space.

The movement of several robots in the same space have a complexity that grows geometrically with the number of robots, making the systematic analysis quite hard. However there are many interesting applications: planetary exploration, automated traffic control, automated warehouses and distribution centres, etc. The level of complexity has also motivated that almost every work related with multiple robot coordination has relayed the coordination effort to some kind of distributed algorithm where each robot takes care of itself and, in those cases that have a higher supervisor, it only considers the general movement of the group and not each individual trajectory.

In relation with the level of coherence between the different individuals movement and their neighbours movement, three different categories can be considered pertaining the movement of several robots in the same space.

- The least coherent movement is that where each individual has its own goal and its movement do not depend at all of the movement of their peers. In this case each robot only takes care of its neighbours in terms of collision avoidance, and do not need to consider the other robots future movements for its own current movement.

This kind of movement is considered as the main scenario on many works from the very beginning of multi robot studies but it is not a closed topic as can be seen in the work of [Guy10], which has quite noticeable results simulating crows interference and coexistence.

- Partially coherent movement is present in those situations where a group of robots moves together, as a group with the same task, but they are not forced to any especial structure among themselves other than stay as a group –keep connected, stay within a distance of the center, remain in visual contact/communication range, etc.–.

This category includes one of the earliest studies of multiple individuals coordinated movement in the work of [Reynolds87], which establishes very simple principles to describe the behaviour and structure of a birds flock. Flocks, herds or schools has been a recurrent research topic since then due to its insight in community and team behaviours both for robotics and biology.

- The more coherent structures for multiple individuals can be found in formations. These gather together individuals with the same final objective while also force them to a specific structure –the classic structure is to keep a given spatial geometry, but the structure ‘locations’ can remain in other parameters different from the spatial position–. Aside from the obvious uses of formations, they represent a very interesting case for study because a formation of individuals can be considered from the outside as a single entity in the same way that a crystal is a collection of atoms with a specific pattern.

While from the side of the robot to keep a formation is the more challenging of the three categories –it needs to consider more elements–, from the side of an external control/task the formation represents, in many occasions, the simplest way. It allows to handle a group as a single individual while keeping the desired constraints.

2.4.1 Formations

Due to its complexity the research has paid a lot of attention to the topic of formation navigation of multiple robots.

Many kinds of formations have been studied, with different approaches on how the robots can be related, how the formation is defined, how do they achieve and keep their shape, how do they communicate, how much communication they do need to maintain the formation, how the formations do perform under stress –sharp manoeuvring– or how they adapt to situations where the formation cannot be kept –navigate around obstacles–.

Many works on formations follow the concept of virtual leaders, where one robot takes the role of the leader and the rest of the robots have to place themselves within a given position related to the leader and then follow the leader movement keeping the relative position. The obvious weakness of these systems was the leader, a failure in the leader or the communication/visibility with the leader means a failure of the complete formation.

Virtual leader approach and some variations of this method can be seen in [Balch95] where the structure is created considering one or many neighbours positions. The use of neighbours positions, takes the initial ideas of the flocking studies and establishes the formations in a way that each robot have to be attached at a given position to one or several other specific robots, these

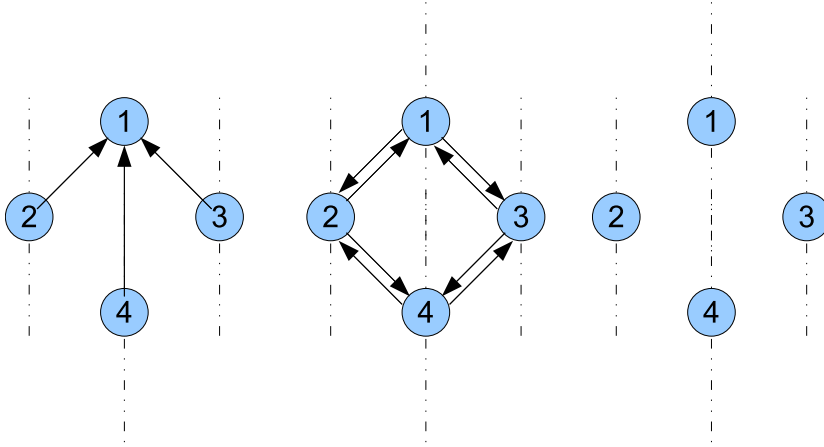


Figure 2.9: Basic Formation Structures: Virtual Leader, Neighbours and Virtual Structure. Each robot is associated with each of the desired locations

robots again are attached to other ones (including the previous ones) and so on. Through this chaining mechanism many formation shapes can be achieved.

An alternative approach to neighbour/leader procedures can be found in [Tan96] in the form of virtual structures. There, a structure with fixed positions for each of its members is defined and then a double control loop works to: a) place the structure optimally over all its members b) move each member towards its given position within the structure.

Along these previous approaches there was a common characteristic: each of the robots has a especial and unique place in the formation structure and those places were not interchangeable. While this can be a desired characteristic in some situations –heterogeneous sensors/satellites/systems arrays– in other situations it is not necessary –homogeneous systems arrays– and results on a loss of agility on the formation initialization, manoeuvring or switching.

The opposite are those formations where each individual can be swapped with any other individual without any difference in the group overall behaviour, since there are not especial roles for the different individuals. These are called anonymous formations and they can be found initially in [Sugihara90], where the principles similar to the ones used in neighbour tracking were adapted to follow characteristics of the geometry, instead of following specific robots by identifying key robots.

Another approach for anonymous formations can be found in [Yun96] where each robot applies a least square function over the position of all its peers and moves in order to reduce the deviation at the next iteration, in this way a line and circle formations are built –a generalization of this method is those scenarios where a team of robots evolves to minimize a communal cost function–.

A third approach to anonymous formations is the use of social fields in [Balch00] where each individual has attractive nodes around itself so other individuals –with similar attracting nodes– attach themselves to one of these nodes

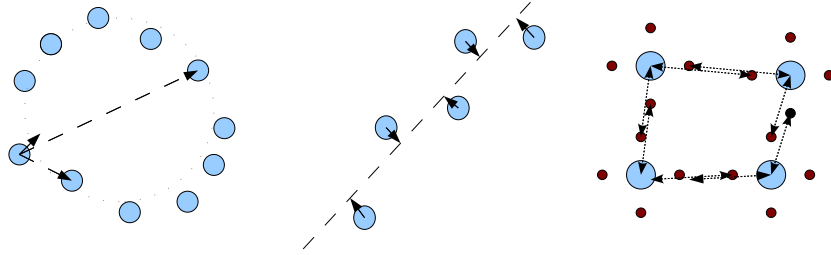


Figure 2.10: Anonymous Formation Structures: Geometry based, Function Minimizing and Social Fields. Each robot have the same role and its position depends on the whole group situation

—and the first robot is also attached to one of the nodes of the second one—. The result is quite similar to a crystallization process.

Other characteristic is how the robot locations are described by the different formation descriptions.

Non-anonymous formations are described in a way that explicitly allocate a position or place for each robot in the formation. This is directly linked with the nature of those formations, were all the robots have an explicit identification and needs an explicit place in the formation or a explicit relationship with other robot.

In the case of anonymous formations, since there is no explicit identification of each robot no explicit relationship with other robot or explicit place in the structure is allocated for each robot. Anonymous formations usually work with relative positions with the neighbours,[Balch00], or relative position within the geometry,[Sugihara90] or [Yun96].

These two approaches have one side consequence: the number of members of non-anonymous formations is fixed, when the number of robots varies, the formation needs to be defined again —this can be done on the fly, but needs to be done—. On the other side, the relative positioning of the anonymous formations is usually open to being re-scalable without any modification on the initial system; in the case of [Balch00] more robots are attached to the free links and the 'crystal' just grows, in the case of [Yun96] more points are considered in the least square function, and in the case of [Sugihara90] more robots must be tested to find the farthest and closest ones, but in general they can handle any number of members.

The main difficulty with fully anonymous formations is how to describe the formation in terms that allow the anonymity of its members, while the formation itself remains flexible enough. This difficulty usually leads to approaches that are non-anonymous but, through explicit communication and coordination, behaves like a fully anonymous formation —robots can freely take any place in the formation—.

Those so called *negotiated* formations include all the variants were there is a leader, but the leader role can be assigned to any robot and it is decided on the fly, or those formations were which robot is attached to witch neighbour,

or which robot is attached which each position of the structure, is reached by consensus among all the robots, like in [Fredslund02] or [Naffin04].

On the negotiation of the formations positions, is notable the auction system as the one described in [Gerkey02]. It provides the means for optimizing the results in semi-heterogeneous systems –several kinds of different systems but several individuals of the same kind– as well as allowing the most versatile of the formation configurations, the virtual structure.

The key on negotiated –and auction– methods is the metric used to establish the cost/value of a specific place on the formation for a specific robot, in order to optimize the cost/value of the complete distribution, especially in non-holonomic robots .

While all the above mentioned structures needed some kind of communication, the non-negotiated ones can usually be achieved through non-explicit communication –knowing were the other robots are, or were are they heading, is a kind of communication, but can be solved through vision or other means that do not involve messages going from one robot to other–, however negotiated formations do require some kind of explicit communication which can not be possible or desired in some scenarios and add an extra layer of complexity to the system –however when there is no problem with explicit communications, negotiated formations can achieve better results–.

It is also worth mentioning the works of [Ge04] and [Kang11] due to their similarity with the formations used as example in this work. Both contributions are a variation of a function minimizing structure, where the function used is a potential field defining a continuous virtual trench to establish the robot positions. In both works the robots must place themselves along the minima region of this trench, however both uses strictly lineal trenches, while in the proposed examples on this Thesis any shape can be set for the trench. In [Ge04] more complex structures are built through the combination of several of these trenches and special role robots that act as trench vertex, which result in a mix of non-anonymous robots –the trench vertices– and anonymous robots –the trench followers–. In the case of [Kang11] the virtual trench only act as a forward wall to avoid the robots to surpass its peers, which limits the results to frontal column formation.

Since formations and multiple robot systems offer a collection of rich test scenarios, the proposed method of information handling and behaviour coordination has been tested in several of these scenarios. The same methodology has been applied to navigate a single robot to a goal, to navigate a multiple unrelated robots moving towards different goals, to navigate a group of robots –linked as a group but without any structure– chasing a goal among some obstacles, and finally to navigate in several formation structures.

Being the formations the richer and most complex cases, they are used for the exhaustive analysis of results in order to validate the proposed method, first in simulation and later with real robots to validate the simulations. The other scenarios are used to illustrate how the proposed method is used, from simpler to more complex behaviour coordination.

The selected formation schemes used for the exhaustive tests are anonymous, can be re-scaled without any modification, they do not have explicit communication –the robots do not negotiate, however they share their position– and the formation structure is a mixture between the traditional virtual structure –with its double coordination loop– and a function minimizing approach using virtual trenches. The formation shapes used for the system testing are the same four found in many articles like [Balch98] : line, column, wedge and circle, since they offer multiple different challenges for the robots.

Just for the completeness, one more formation scheme has been added: classical virtual structures with specific locations for the robots –still using anonymous robots–. There is no intensive testing on this formation *shape* since it is built as a particular case of the trenches formation, however this kind of formation is quite useful in many situations and it is frequently visited in literature. Showing this kind of formation rounds up the collection of examples for the proposed method by showing all the classical multiple robot structures.

2.5 Custom related works

Along the time on which the work presented on this Thesis has been developed, some of its elements were presented and published. These works show how the base ideas evolve and all the different elements were put together, ending in this Thesis.

In these previous works, [Cifuentes06] introduced the simulation environment, in [Cifuentes08a] and [Cifuentes08b] the use of the potential fields for handling groups of robots is introduced, [Cifuentes10] introduces the application of the method to formation structures, and finally a complete overview can be found in [Cifuentes12a] and [Cifuentes12b] which present a more detailed analysis of both the abstractions and the application of the method.

At Appendix C it can be found how these works were developed, showing the different elements as they evolved along time.

Chapter 3

General overview

In the previous chapter it has been shown that there are many contributions about different aspects of robot navigation, from individual robot movement to highly structured formation patterns of moving robots. Usually these works are centred around an specific aspect of the navigation or aimed to a specific task. The solutions provided by those works are usually bent in the direction of an specific aspect or task, being difficult to find the boundary between the pure navigation scheme and the task itself.

While all these works can be of the highest quality and have astounding results, some kind of generalization on the method is sometimes missed. Which are the elements needed for the navigation itself, taking out the specifics of the case?. What kind of dependence have that specific case on the navigation? What is the minimum information –not the optimal– that is needed to solve a given case?. And above all, is there a systematic way to approach any given case through the proposed methodology?. How can been identified the minimum dependent elements of the specific case and how apply them following a systematic way?

Along this work there is an attempt to identify the specifics of the methodology and their systematic application on increasingly complex scenarios; dealing with the different definitions of the local navigation for each scenario in a systematic way.

3.1 Initial approach

As in many other cases, this Thesis began with the study of a slightly different objective: a way to handle one or many robots with different relationships as a single entity.

Along that initial research, arose the necessity to identify which elements were common to the different studied cases, which elements were specific of each case and how that specific elements affect the navigation process.

At that point the initial research evolved on the study about how to deal with many different relationships among the robots in a systematic way.

An extensive analysis was done then, to separate those elements that cover the navigation part of the problem from those elements specific to the case where the navigation is applied.

After this classification, the initial objective of being able to handle all the robots as a single entity was still considered, but actually, handling all the robots as a single entity affects very little to the navigation scheme: it was found that it can be considered a case specific element.

After decomposing how the different authors handle the navigation of their robots, a general idea was reached about the elements needed for the navigation and those related to the specifics of the case .

Most of the studied contributions express their methods in terms of the behavioural approach, since the decomposition of the main task in multiple individual subtasks made easier to handle the complexity involved in the main task. However when the methods were implemented, the individual subtasks were not always kept independent one from each other, but when the actions must be applied over the actuators they become entangled, due to the precedence rule analysis or to the self-balancing of the different terms. Therefore, if those implementations were going to be adapted for different uses, most of them will need a complete reimplementaion.

The difficulty remains in how to express the terms of a specific case in a way that do not directly affect the basic needs of the navigational process and in a way that was able to keep the different considerations of the case, the different subtasks, apart one from each other. The classical approach of working directly on the actuators space from the case specific elements does not provide the kind of results that were being searched, since that approach was in the center of the entanglement.

3.2 Main Proposal

Along the next chapter, a method is proposed to keep independent the different considerations needed to guide a robot, independent one from each other and independent from the actuation space.

The proposed method have two well differentiated elements: How to express the actions needed for each independent consideration –from each behaviour– and how to apply these needed actions into the actuators.

3.2.1 Action blending

The main characteristic of the proposed method is that the dimensions of the action space –the results of the logic evaluation– and the actuator space –the physical capabilities of the robot– are not the same, the actions space has higher dimensionality than the actuators space.

The objective of this extra dimensionality of the actions space is to increase the information handled by the system, thus providing a way to avoid conflicts which would have arisen if the actions would have been expressed in the actuator space, and so increasing the easiness on how to express the desired robot behaviour.

In this work, the way to avoid conflicting actions is to keep some kind of background motivation to the actions, and through this background motivation ease the decision process when actions are blended to be applied to the robot actuators.

A very simple example can be used to illustrate this by following the reasoning taking place in a robot which moves to its goal and finds an obstacle in its way.

- When the actions space and the actuators space is the same, there is a conflict as can be seen in figure 3.1

The goal-seeking subtask tells the robot to move forward, because the goal is directly ahead. And, at the same time, the obstacle-avoidance subtask tells the robot to move backwards –or to move in any direction but ahead–, since the obstacle is directly ahead.

The two subtasks provide conflicting solutions but the two actions are perfectly legit. Since the background of such reasoning is not transmitted in any way and, when the robot have to apply these actions to the actuators, there is no way to discern which one is the good one.

– A quick solution will be to establish a priority of one subtask over the other one, but that leads to the entanglement observed in other works. The hierarchical structure was the *subsumption* approach from Arkin, but it was proven that the hierarchical approach only works for simple cases. When the complexity of the cases grows, a mechanism to dynamically establish the priorities is needed, and that mechanism is specific for each subtasks structure –

- The approach used along this Thesis increases the dimensionality of the actions for a same actuator set. We can expand the *move* actuation into two actions also addressed to move the robot: *try-to-move* and *forbid-the-movement*.

Considering these two actions is straightforward how to apply them to the considered situation. The goal-seeking subtask will instruct the robot to *try-to-move* ahead, while the obstacle-avoidance subtasks will instruct the robot to *forbid-the-movement* ahead.

When all the subtasks were evaluated and the moment to apply some movement is reached, the decision process will be much easier –the *forbid-to-move* action have priority over *try-to-move*–, since we have some perspective about the actions that the different subtasks have established.

– In some way, this approach can be seen as establishing a priority scheme among the different dimensions of the action space, but the difference with the *subsumption* approach will be that the priority scheme is not related with the used subtasks so it can be applied to any subtasks structure –

The extra dimensionality of the action space over the actuator space is determined by the problem itself and the requirements of the task. The relationship between the different dimensions of the actions space, for its merging into the actuators space, is only defined by convenience on how to express the actions themselves.

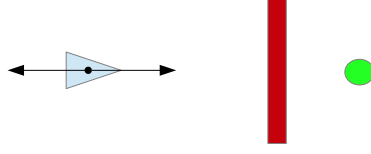


Figure 3.1: Single Robot with obstacle ahead. Classic approach with single valued actions applied directly on the actuators

For this Thesis, being the application field the robot navigation, the increase of the dimensionality of the *move* actuation into the *try-to-move* and *forbid-the-move* actions –where the *forbid-the-move* actions modulates the *try-to-move* actions– is enough to dramatically improve the easiness on how the navigation of a robot is defined.

Just eight actions are needed to guide the robots inside a general formation structure and keep the formation structure stable; obstacle avoidance and collision avoidance with other robots is done through just other five actions.

3.2.2 Action description

A secondary, but also important element of the proposed method is how the actions are specified.

The frequent approach of setting the action/actuators using single values implies a quite reduced range of valid options to be taken at the decision point where all the actions of the different subtasks are considered. On the other hand, when actions/actuators can be defined through multivalued statements, the range of valid options to be taken at the decision point is increased without adding much complexity to the expression of such actions/actuators.

Taking again the previous example of a robot moving toward the goal which finds an obstacle in its way we can illustrate this:

- When the actions *try-to-move* ahead and *forbid-the-move* ahead are set as single valued actions, at the decision point, the robot will apply the precedence of the *forbid-the-move* action over the *try-to-move* action. If no other action is stated, the robot will not move at all since no viable movement direction have been provided.
- Those two same actions can also be expressed through multivalued statements as depicted in figure 3.2.

The *forbid-the-move* action is stated to cover all the obstacle surface The *try-to-move* can be stated centred in the goal direction but also including all the directions surrounding the goal –with decreasing preference– for $goal \pm \pi/2$.

Using multivalued actions, at the moment of the decision process to merge all the subtasks, the robot will have still available all the directions not directly covered by the obstacle.

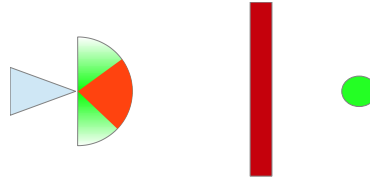


Figure 3.2: Single Robot with obstacle ahead. Proposed approach with multi-valued actions with an increased actions space

3.2.3 World modelling

The two previous elements defines how the decisions on the robot movement are going to be stated and how the different statements are going to be put together to finally move the robot.

However, in order to take any decision, the robot must evaluate the world in which it moves, so the world needs to be described in a way that can be analysed by the different subtasks.

Being the application area of the Thesis the robot navigation, the main element needed to evaluate the robot situation and set the different actions is the relative position of the different objects in the world.

Evaluating each subtask against each of the elements in the world of the robot is a valid approach and do not conflicts with any of the previous considerations. However, as the world grew in complexity, the computational effort to evaluate all the elements in the world will quickly increase, but only a few of those elements will be actually relevant for the robot navigation.

To reduce the computational load of evaluating all the subtasks against each element on the robot world, multiple virtual potential fields are used in the proposed method to condense the information of the world.

By merging all the information of the world using the virtual potential fields, a single evaluation of all the different subtasks is needed to completely evaluate the world, independently of the number of elements placed into it.

Multiple virtual fields are built, instead of a single one, to keep isolated the different characteristics needed to be evaluated by the different subtasks; i.e. the subtask taking care of the obstacle avoidance uses a virtual field built only from obstacles and which does not contain any contribution from the goal or robots in the world.

Each of the virtual potential fields is built to reflect the influence of the most relevant elements needed to properly evaluate the different subtasks; i.e. the virtual potential field used for the obstacles will reflect the position of the nearest obstacles, while the position of the farthest ones will be barely reflected.

Using those three main elements –high dimensionality action space, multivalued action specification and the use of multiple virtual potential fields– a full method can be proposed and developed to guide the navigation of a robot.

3.3 Test Cases

The selected cases to test the proposed method have been chosen to incrementally show the method capabilities while the scenario complexity arises as can be seen in the preview images at figures 3.3 to 3.6.

The first test case is the already described scenario of a single robot moving towards the goal and finding an obstacle in its path.

This first case illustrates how the robot situation can be evaluated and how to state the actions needed to safely guide the robot to the goal. It also shows how the main problem of the potential fields, the local minima, do not appear when using the proposed approach.

The second test case involves the navigation of multiple robots in the same world, all of them guided using the proposed algorithm. In this case it is shown how the first test case can be expanded to include new considerations –the presence of other robots– and how this new consideration barely has influence with the ones designed for the single robot scenario. Also, along this second test it is shown how more challenging conflicts among sub-tasks are handled by the proposed method.

The third scenario increases even more the complexity of the subtasks structure –but the world complexity remains equal as in the second studied cases– and illustrates how the information of the world can be gathered in different potential fields for different purposes –using the same potential sources, gathered in different ways, to handle two different objectives–.

The fourth tested scenario shows how to apply the proposed method to handle the navigation of a group of robots in any given formation shape with just a few simple actions.

Through this test case the blending capabilities of the proposed method for complex tasks is being demonstrated.

To support this hypothesis, an extensive work using different number of robots, different formation shapes and multiple tests using random initial conditions of the scenario are done.

Also, since all the previous test cases are illustrated only through simulation, a reduced –but still significant– subset of tests is done using real robots to both validate the simulation results and show the applicability of the proposed method to real robots.

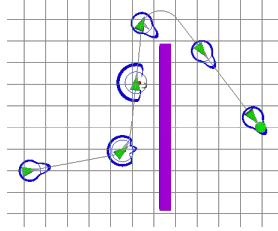


Figure 3.3: Preview of the single robot scenario.

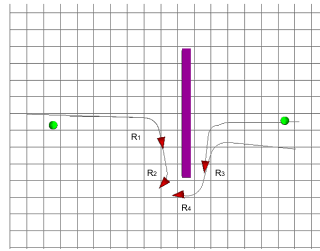


Figure 3.4: Preview of the multiple of robots scenario.

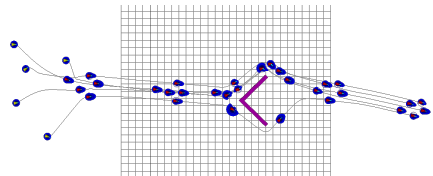


Figure 3.5: Preview of the unstructured group of robots scenario.

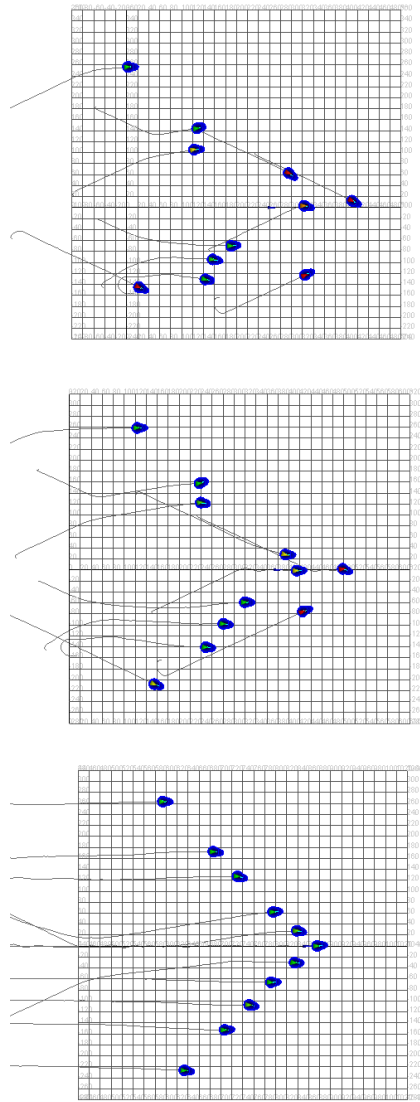


Figure 3.6: Sample of the process to reach a formation structure -wedge- using 11 robots

3.4 Results

The first test cases are used to illustrate the how the proposed method work, and their results are just punctual indications of the method evolution. However, the last test case, formation navigation, being deeply analysed, provides a good support on the capabilities of the proposed method.

After the simulation of the formation navigation for different number of robots, it is observed how the evolution of the distance to the formation for each robot depends mainly in the number of robots and this dependence is linear, while is only secondarily influenced by the formation shape. While this result is more or less intuitive, if the method were not able to solve complex situations, the dependence will not be linear, because the complexity of the complete scenario grows geometrically with the number of robots.

This linear dependence with the number of robots is observed along all the statistical variables analysed, thus supporting the hypothesis that the behavioural approach can distribute the complete scenario complexity among its members and that the proposed method is able to handle the increasing complexity without increasing effort.

Chapter 4

Method Description

4.1 Mobile Robots

The mobile robots considered for this work are non-holonomic vehicles. The limitations chosen for the movement of the robots are those that mimic the limitations of common vehicles, such as cars or simple boats. The robots can move forward, and, once moving, can apply some rate of turning for manoeuvring. No turning can be done if the robot is not moving and the turning rate is constrained by the forward speed.

Robot restrictions In general forward speed can take any value between 0 and ν_{MAX} and the angular velocity of the robot is limited according to eq. 4.1. The minimum turning radius is r_{Lim} . Taking $r_{Lim} > 0$ implies that the robots are not allowed to spin.

$$|\dot{\theta}| \leq |\nu|/r_{Lim} \quad (4.1)$$

Heading and speed keeping The robots are supposed to have their own motion control loops for keeping speed and steering directions. The local navigation method provides to the robot the desired speed and steering reference values. During the research, inertia, and slip effects are not specifically considered, however noise effects on the robot movement are included in simulation.

For the proposed method, each robot needs to know the position and orientation of every other robot on the formation. How this knowledge is acquired by the formation members is not a topic of research here; several methods already exist and each one have applications on different situations. Also the relative positioning of the obstacles, or at least the minimum between the robot and the obstacles, is needed by the robot when it is located in the vicinity of an obstacle. Again the measurement method is not discussed along the current work.

Robot parameters In order to provide a reference, along the simulations done for this research the values of r_{Lim} and ν_{MAX} are taken from the ones of the experimental robots, which are $r_{Lim} = 0.1m$ and $\nu_{MAX} = 0.1m/s$.

4.2 Algorithm Elements

The proposed algorithm makes use of several structures and constructions. These elements are mainly used to represent the data managed by the algorithm. While they are not strictly necessary, these constructions make easier to understand the whole process of the algorithm.

These structures are used to gather different sets of information managed by the algorithm: the robot scenario, represented by means of *virtual fields*; the possible situation that can be found by the robot, represented by *situation descriptors* and the actions that can be taken by the robot in each situation, represented by *action maps*.

4.2.1 Virtual Fields

Virtual fields and virtual forces have been extensively used in robotics since their introduction by Khatib in [Khatib78]. They provide a simple and intuitive way to represent and visualize situations otherwise too complex to be represented due to the presence of multiple elements affecting a large area.

The virtual fields commonly used are compositions of central force fields, which can be expressed as negative gradient of a potential. In equations 4.2 and 4.3 r represents the distance to the field source i , and a_i and p_i are the parameters that shape the field.

$$\phi_i(r) = a_i r^{p_i} \rightarrow \vec{f}_i(r) = -\nabla \phi_i(r) \quad (4.2)$$

$$\Phi = \sum_i \phi_i \rightarrow \vec{F} = \sum_i \vec{f}_i = \sum_i -\nabla \phi_i \quad (4.3)$$

Classical Virtual Fields When several sources are present, as hinted in eq. 4.2 and eq. 4.3, the virtual field is built by adding together the effects of each field source along the scenario area. While this is the classical approach when working with virtual fields, it presents some limitations and problems well described in literature. Of these problems, the most common one is the surge of one or more field local minima on the scenario. Since the navigation algorithms usually employed with virtual fields are variations of a gradient descent algorithm, the presence of local minima can result on a robot trapped inside a local minimum. Potential field navigation is usually a reactive navigation method. The robot does not plan in long term its movement, only reacts to the instant measures of the field at its current location, therefore when the robot is following the field gradient it is unable to discern if it is descending into a local minimum or into the goal minimum. Figure 4.1 show a simple scenario and the resulting field which have two minima. One of the minimum is due to the goal, which have an attractive field attached. The other minimum is due to the overlapping of this goal attractive field with the repulsive field of the obstacle. In both minima the field gradient at the minimum point is null, $\vec{F} = 0$, and in both the potential is non zero, $\Phi \neq 0$, that can be best appreciated in figure 4.2.

Proposed Virtual Fields In the presented method, in order to reduce the influence of local minima and other problems, the employed virtual field does

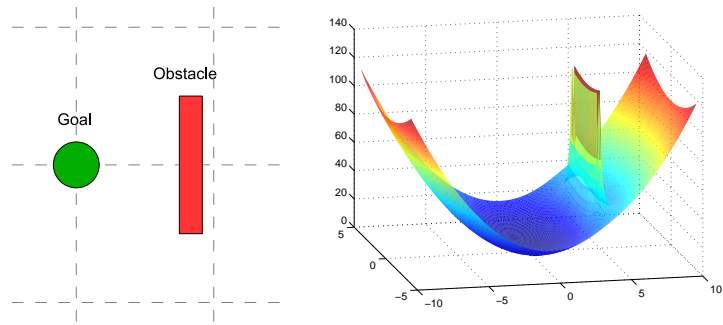


Figure 4.1: Goal and single obstacle scenario with its resulting field

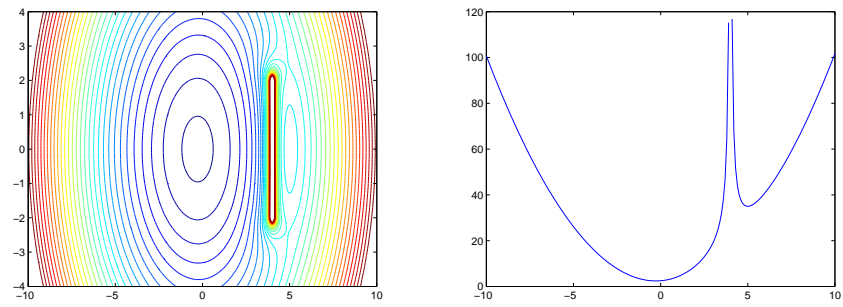


Figure 4.2: Contour and side cut of the field value for a simple scenario with a goal and an obstacle

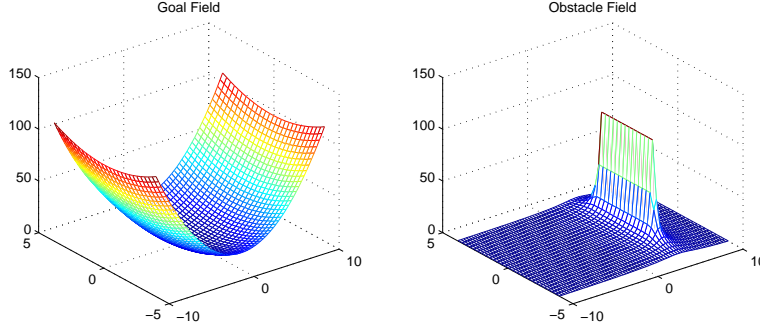


Figure 4.3: Goal and Obstacle fields surface keep independent

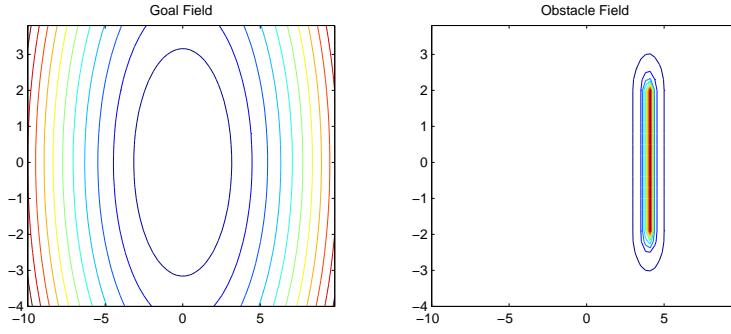


Figure 4.4: Goal and Obstacle fields contours keep independent

not unify all the present field sources. Instead, the different sources are classified in different kinds, and only those of the same kind are unified. The result is a collection of fields where each field represents the *information* from a specific kind of scenario element. Figures 4.3 and 4.4 shows the same scenario as in 4.1 but now the fields associated to the goal and the obstacle have been keep apart, and they will be sensed independently by the robot.

The criterion used to separate the different kinds of field sources is to avoid mixing sources that have different nature –attractors and repulsors– or that needs to be considered in different ways –obstacles and robots–. The field sources that can be treated in the same way and can be considered as a single *distributed* entity can be unified: all the obstacles together or all the robots together, but not a mix of obstacles and robots.

The traditional field can still be obtained because when a robot builds the fields grouped by kinds, the addition of these fields will result in the same unified field, Φ , of the classical approach shown in equation 4.4.

$$\Phi = \Phi_{goal} + \Phi_{obstacles} + \Phi_{robots} + \dots \quad (4.4)$$

The classical methods and their advantages are still applicable, because the single unified field can still be built. Having the sources grouped in *kinds* in-

creates the level of information managed by the robot while keeps the amount of available information into reasonable levels. This grouping of information allows the robot to discern more accurately its situation but avoids the large amount of information that will result from considering each field source individually. Through the use of grouped fields is trivial to identify and avoid the problem previously described for the unified field: the local minimum due to the goal attractive field or the local minimum due to the overlap between the goal and obstacle fields.

Employed field sources The goal and obstacle potentials chosen for our present work follow the classic expressions for attractive and repulsive field shown in equations 4.5 and 4.6

- *Attraction potential caused by a goal*

$$\phi_{GOAL_i}(d_i) = a_i d_i^2 \quad (4.5)$$

- *Repulsion potential caused by an obstacle*

$$\phi_{OBS_i}(d_i) = b_i / d_i^2 \quad (4.6)$$

Since the goal and obstacle fields are not added together there is no need of fine tuning the parameters in eq. 4.5 and eq. 4.6, a_i and b_i , to shape the behaviour of the robot. Usually these parameters are used to determine the maximum allowed proximity between the robot and the obstacles, where each obstacle needs a different value due to the presence of other field sources. However when all the obstacles are considered as a single entity and the obstacle field is not mixed with any other fields, this parameter can be the same for all the obstacles. Along this work these parameters will always be 1.

4.2.2 Perception and data handling

The specific detail on robot capabilities: sensors and locomotion, is not considered for the algorithm construction. It is supposed that all sensor data are used for building the potential fields. The built potential fields are the starting point for this approach.

Along the navigation the robot will face different circumstances which will require some action, we will refer to each of these circumstances as a *Situation*, e.g. to have an obstacle in the path to the goal. Those *Situations* are specified by combining one or more elements that will be named *Descriptive Elements*.

The construction of *Descriptive Elements* out of field data and *Situations* out of *Descriptive Elements* is done in the same way as fuzzy sets are built. A specific membership function, applied over the field data, transforms the raw data into a more human understandable *Descriptive Element* and the full flagged *Situation* is built by combining, through fuzzy logic operations, one or several *Descriptive Elements*

Descriptive Elements

The *Descriptive Elements* are entities directly built using the information extracted from the potentials and virtual fields which are normalized as continuous

values (0..1) or (-1..1). While *Situations* can be built directly out of field information, the use of the descriptive elements simplifies the process of describing a complex situation by using smaller elements that are expressed as a linguistic concept instead of its full mathematical expression; this allows to better understand what factors must be considered and how a *Situation* can evolve. The value associated with a *Descriptive Element* expresses the level of accomplishment of the atomic situation that it describes and also the value associated with a full *Situation* built through *Descriptive Elements* express the level of accomplishment of the described situation.

Descriptive elements combinations The normal operations performed usually on descriptive elements will be negation and combination operations. Since the descriptive elements are normalized as continuous functions between 0 and 1 the intersection operation, or logic *AND*, can be easily done through a product operation between elements (eq. 4.7). Also the negation of a descriptive element can be done through subtraction from the unit.(eq. 4.8) Having described intersection and negation operations a full logic system can be built while keeping continuity and normalization of the operators. One more operation between descriptors is applied along the construction of the situations. In order to shift the relevance of one operator when is combined with another one, a power operation can be performed. This power operation do not disrupts the normalization or the continuity but allows to emphasize the desired value, towards 0 or towards 1, of the descriptor in the building of a situation, e.g. transform "near an obstacle" to "very near to an obstacle". In general it will be something like eq. 4.9 where the results can be seen in figure 4.5, there it can be appreciated that the DE^2 measurement will need lower raw data values to match DE result, while the inflexion points and general normalization remains equal.

For consistency reasons all elements considered in a situation must happen at the same time.

$$DE_1 \& DE_2 = DE_1 \times DE_2 \quad (4.7)$$

$$!DE_1 = 1 - DE_1 \quad (4.8)$$

$$"Very" DE = DE^2 \quad (4.9)$$

Descriptive Element Examples The two main magnitudes that can be obtained from the potential fields are distances from and relative orientation of the fields sources to the robot, therefore the *Descriptive Elements* employed refer to distances and orientations of the elements in the scenario.

The *Descriptive Elements* related to the distances measurement are built using a sigmoid function. These *Descriptive Elements* will represent when two elements are near, far or in-between a region of especial interest for a given situation. The employed sigmoid function, eq. (4.10), defines the interest region through the values x_{001} , which establishes the point where the sigmoid takes the value of 0.01, and x_{099} , which sets where the sigmoid goes to 0.99. In this way the characterization of the sigmoid function results very intuitive.

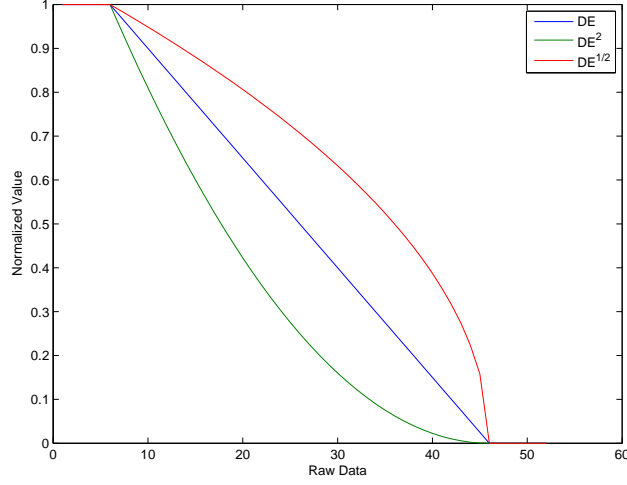


Figure 4.5: Descriptive Element and the emphasizing result of the power operation

$$\begin{aligned}
 Sg(x, x_{001}, x_{099}) &= \frac{1}{1 + e^{-(x-o) \cdot s}} \\
 o &= (x_{001} + x_{099}) / 2 \\
 s &= (\ln(\frac{1}{0.01} - 1) - \ln(\frac{1}{0.99} - 1)) / (x_{099} - x_{001})
 \end{aligned} \tag{4.10}$$

Using this function it is possible to define a *Descriptive Element* example: *Obstacle near* describing when an obstacle is near the robot, so the robot can take care of it. In equation 4.11, the values d_{Far} and d_{Close} are values defined in the logic system of the robot and $Obs_{Distance}$ is measured indirectly from the obstacle field. For a $d_{Far} = 50cm$ and $d_{Close} = 20cm$ the result can be seen in Figure 4.6

$$Obs_{Near} = Sg(Obs_{Distance}, d_{Far}, d_{Close}) \tag{4.11}$$

Descriptive Elements based on angular measurements takes their raw data from the different field gradient orientations at the robot position, therefore the angular values are always relative to the current robot orientation. In some cases, these measurements can be handled through a linear transformation, however other transformations can also be applied to better shape the region of interest. Lets consider a very simple case: a descriptive element, *Obstacle Ahead* that signals when there is an obstacle in front of the robot; when the obstacle is directly ahead, the obstacle field gradient orientation, θ_{Obs} , will be $\pm\pi$ –the obstacle field is repulsive–. A linear normalization like in eq. 4.12 will provide the desired result when the obstacle is strictly ahead, however a non linear normalization like the one in eq. 4.13 provides higher values around the critical point of $\pm\pi$ and lower values once the obstacle is at the back of the robot, $\theta < \pm\pi/2$. These can be appreciated in in Figure 4.7 and its polar representation, Figure 4.8.

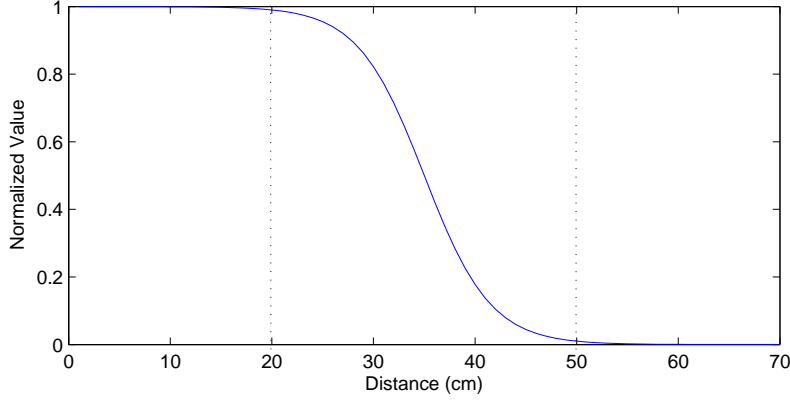


Figure 4.6: *Obstacle Near* : $d_{Near} = x_{099} = 20cm$, $d_{Far} = x_{001} = 50cm$

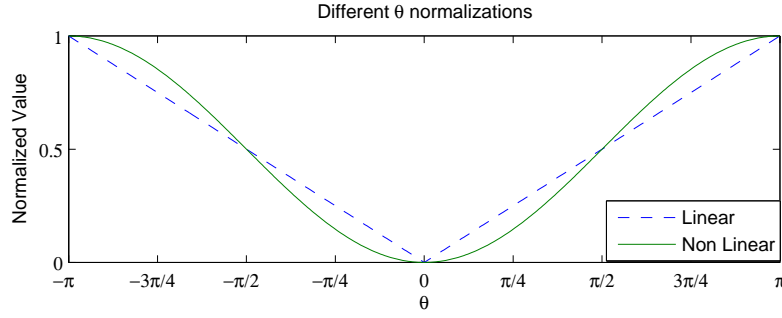


Figure 4.7: Linear and a Non-Linear normalizations of an angular measure

Normalizations like the one in eq 4.13 are usually employed along this work. Other normalization functions can also be valid and provide similar results. This kind of normalization has been chosen because it provides good enough results while the mathematical expression remains simple enough.

$$\left| \frac{\theta_{Obs}}{\pi} \right| \quad (4.12)$$

$$Obs_{Ahead} = \frac{1 - \cos(\theta_{Obs})}{2} \quad (4.13)$$

Situations

A *Situation*, as used in this work, represents a collection of perceptions that fulfills a specific relationship, e.g. to have an obstacle ahead and near enough. Each individual *Situation* have associated a set of actions, intended to help the

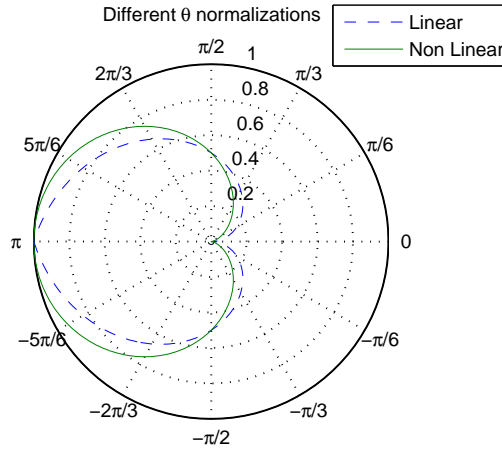


Figure 4.8: Linear and a Non-Linear normalizations of an angular measure, polar representation

robot to deal with the situation, e.g avoid the obstacle. Therefore the different actions taken by the robot will depend on the evaluation of the *Situations* perceived by the robot. The level of accomplishment of a *Situation*, its value, is obtained by a specific combination of operations on the *Descriptive Elements* that compose the situation. The resulting measurement of a *Situation* will be again a normalized continuous value (0..1) and the effects of actions associated with a given situation will be weighted by the *Situation* measurement.

An example *Situation* can be built by considering the *Descriptive Elements* Obstacle Near, eq. 4.11, and Obstacle Ahead, eq. 4.13, described above. We can define a sample situation, *Collision Danger*, evaluating the direction of the obstacle and its distance to the robot by their *Descriptive Elements* product, eq. 4.14.

When the obstacle is in front of the robot and the distance is near enough to start the avoiding maneuver, the situation evaluation will reach the full value, 1. When the obstacle is not directly ahead or it is farther than d_{Close} the value associated to the *Situation* will decrease. A representation of the *Collision Danger* results can be found in Figure 4.9 for obstacle positions surrounding a robot located placed at (0,0) and with $\theta = 0$ –pointing to the right–. The intensity of the action associated to this situation, avoid the obstacle, will be associated with the situation measurement.

$$S_{CollisionDanger} = Obs_{Near} \times Obs_{Ahead} \quad (4.14)$$

In the proposed navigation method multiple situations, with their actions, will be considered to define the robot behaviour. All the situations are evaluated in each control cycle and all the actions are considered in each control cycle. However the actions will be weighted by their associated situation measurement, so only the actions associated with the more relevant situations will actually be significant.

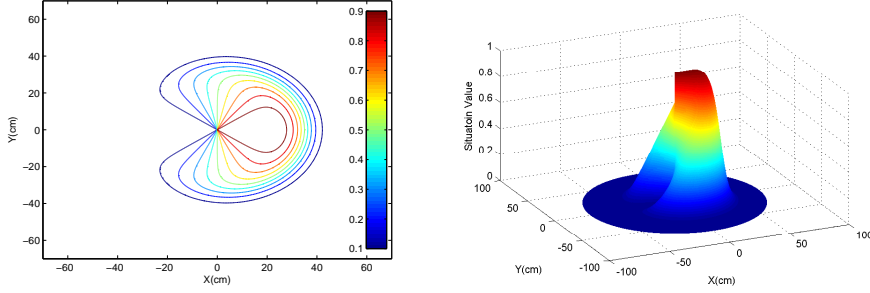


Figure 4.9: Evaluation of $S_{CollisionDanger}$ for different relative obstacle positions

Multiple field sources data The previously described examples of *Descriptive Elements* and *Situations* refer to distances and angles to a single obstacle. However the most common situation will be that many obstacles were present at a given scenario. The robot will not use the information of each obstacle to obtain all distances and angle but each one of these obstacles will add a contribution to the obstacle field sensed by the robot.

The use of the composed field allows us to extract a single distance and angle value that will be equivalent to a single obstacle which sources the same field as the one sensed by the robot, eq. 4.15. When there is only a single real obstacle in the scenario, the equivalent obstacle and the real one will be placed at the same location, in other situations the expression of the repulsive field will motivate that obstacles closer to the robot will be the most influential ones while farther obstacles will barely influence the virtual obstacle location, this is depicted in figure 4.10.

The distance and angle of the equivalent obstacle will be the ones used to build the *Descriptive Elements* and *Situations*. These values are obtained by reversing the repulsive field equation introduced in eq. 4.6 and its gradient while considering that all the field is sourced by a single obstacle, the equivalent obstacle, which results in eq. 4.16 and eq. 4.17.

When the distance to the virtual obstacle is extracted from the potential

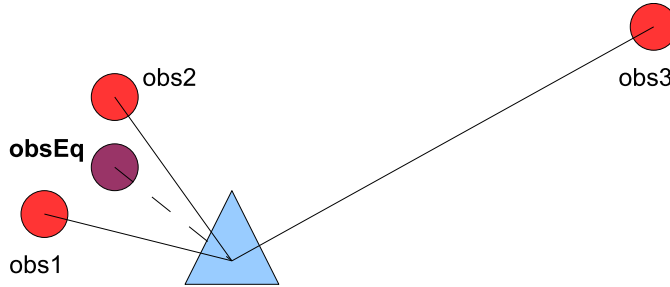


Figure 4.10: Equivalent obstacle sensed by the robot when several real obstacles are present in the scenario

field data, due to the vectorial nature of the field and the non-vectorial nature of the potential two different distance measures can be obtained, one from the field gradient module and one from the potential value. These two values will be equal only when all the obstacles are aligned at the same side of the robot, when not, the distance value obtained from the field gradient module will show a greater distance measurement –lower module– than the distance value obtained using the potential value.

The distance value employed for the construction of the *Descriptive elements* is usually chosen as the more conservative, safer, one. For example in the collision distance measurement the distance obtained from the potential value will be employed since it will return the smaller value for the distance, thus the collision avoidance manoeuvres will start earlier.

The disparity between the distance value obtained from the field gradient module and the potential value is also used in some cases to establish when a robot is surrounded by obstacles.

The aggregation of field sources of the same kind and construction of a single equivalent source, here detailed for obstacles, is also applied to all the different kinds that have multiple objects times in the scenario.

$$\Phi_{Obs} = \phi_{obs0} + \phi_{1\ obs1} + \dots + \phi_{obsN} = \phi_{ObsEq} \quad (4.15)$$

$$d_{ObsEq}(\Phi_{Obs}) = \frac{1}{\sqrt{\Phi_{Obs}}} \quad (4.16)$$

$$d_{ObsEq}(\|F_{Obs}\|) = \frac{2}{\sqrt[3]{\|F_{Obs}\|}} \quad (4.17)$$

4.3 Decision process

The final objective of the situation evaluation and their associated actions is to guide the robot along a task. Usually the situations will be built independently to deal with very specific circumstances, but frequently, more than one *situation* will present a significant value.

In general it should be considered that several situations will coexist, and that the action associated with each situation is crafted to work alone. Since all considered situations will be simultaneously evaluated, many actions will be suggested at each cycle. Those actions must be processed to establish the single final direction and speed for the robot. The decision process is the responsible of taking all the actions suggested by all the relevant situations and analyse which final action will fit best with all the suggested actions.

4.3.1 Action Description

In many cases a given situation can be solved in more than one way, the solution is not unique, moreover a broad set of actions will be equally good to solve the situation; e.g. when an obstacle is directly ahead, to skirt it by the right is as good as to skirt it by the left and, from the strict obstacle avoidance point of view, to turn $\pm\pi/2$ from the obstacle direction is as good as to turn $\pm3\pi/2$.

The description of the actions associated to the situations should, therefore, allow to represent these broadness of equally good solutions for a situation. We have also stated that several situations may coexist, then, the different actions associated with these situations will need to be evaluated together.

The most classical and simple approach to the evaluation of multiple independent solutions in mobile robotics is the vector addition. There, each solution is represented by a vector with a given action direction and a given module that weights the solution relevance. The different vectors provided by the different sub tasks are simply added together and the final result is directly used as the movement/action vector to be applied to the robot.

The vector sum techniques, while proven good enough for simple situations, do not present a good performance when the amount of vectors increases, requiring of expert fine tuning and additional rules to work. Also due to their vectorial nature they are not good to represent the broadness of the solutions desired in this work.

Instead of using vectors to represent the actions that can be taken by the robot, the actions in this work are going to be expressed in form of a histogram that covers the complete circumference around the robot in a similar way that VFH methods works. However we represent actions to be taken instead of occupied spaces around the robot. Using a histogram around the robot circumference allow us to express actions with broad application area and multiple simultaneous actions can be gathered semi-independently.

From here on, a single action should be considered as of a modified bell-function which maximum will state the desired relevance of that action, the wideness of the maximum expresses the contiguous set of directions with equal relevance and the side slopes represent how the relevance of the solution decays for the directions which are not in the maximum area. In the case of situations with more than one solution, the histogram holds multiple single actions so multiple regions can show a maximum.

To express all the individual actions in the same easy way, we will employ the modified bell-function found in eq. 4.18.

$$\tilde{\psi}(\theta; \mu_i, \sigma_i, \delta_i, w_i) = \begin{cases} \psi(\theta; \mu_i - \delta_i, \sigma_i) * w_i, & \mu_i - \pi \leq \theta \leq \mu_i - \delta_i \\ w_i, & \mu_i - \delta_i < \theta < \mu_i + \delta_i \\ \psi(\theta; \mu_i + \delta_i, \sigma_i) * w_i, & \mu_i + \pi \geq \theta \geq \mu_i + \delta_i \end{cases} \quad (4.18)$$

$$\psi(\theta; \mu, \sigma) = e^{-\frac{(\theta - \mu)^2}{2\sigma^2}} \quad (4.19)$$

Where $\psi(\theta; \mu, \sigma)$ represents the bell curve at eq. 4.19 in which the angle θ is measured respect to the robot current heading.

The function $\tilde{\psi}(\theta; \mu, \sigma, \delta, w)$ is a widened and weighted version of a bell function, where δ represents the expansion of the top value of the bell and w is the scale factor. In the case $\delta = 0, w = 1$ this function will be equal to the bell-function $\psi(\theta; \mu, \sigma)$.

Figures 4.11 and 4.12 show the classic bell curve and the expanded bell curve that are used to define the actions.

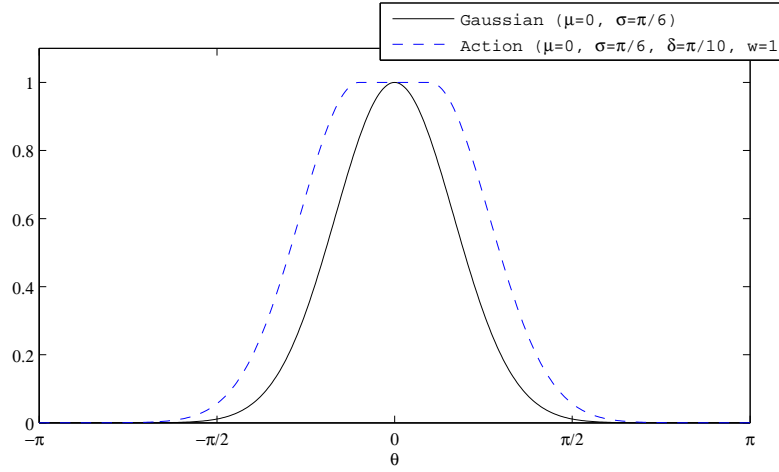


Figure 4.11: Classic Gaussian Curve and Extended Gaussian Curve used to define the actions

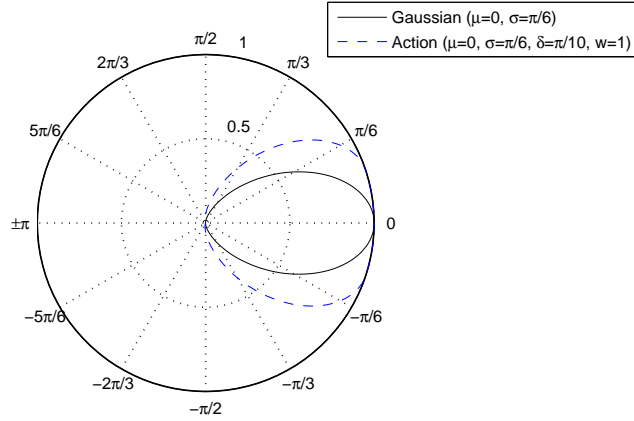


Figure 4.12: Polar representation of the classic Gaussian Curve and Extended Gaussian Curve used to define the actions

The use of a bell function to express the actions associated to the situations is just a convenience. The normal bell curves provide an easy way to designate a preferred value while allows also to specify how the values around the preferred one should be considered. The added extension term, δ , provides the way to designate a preferred region when necessary, instead of a preferred single point. While other kind of functions can be used for this same purpose, the bell function have been chosen because of its common use and well known properties.

4.3.2 Employed Actions

It is quite common that the actions taken through the evaluation of the robot and environment state were represented just by the desired movement direction and/or speed for the robot that handles the given situation. This process is quite straightforward and gives proper results but, as the number of actions to consider rises, and specially as the amount of actions that may coexist is increased, this method is not enough. The problem is how to discern among actions with similar relevance but different, even opposite, desired movements associated; there is not enough information about why each action should be taken.

To reduce this problem the amount of information that supports the handling of each situation has been increased to ease the final decision process. Instead of just using the pair, direction and speed, each situation can state three kinds of actions:

- *Elect*: For the desired robot movement, a set of directions can be defined, at any given situation. These eligible directions represent those movement directions which are suitable for the situation to come.
- *Forbid*: Also a set of directions can be defined for those movement directions that not only do not fit with the solution of the current situations but that leads to a specific threat to the robot. The forbidden actions act as a veto over the elected actions, so in the decision process the forbidden directions are going to be considered as something to avoid, no matter how much level of *Eligibility* they have associated.
- *Brake*: To control the speed for any given direction a histogram that represents speed restrains that the robot should apply based on the knowledge contained on a given situation. This provides a mechanism to limit the speed for any direction associated with a certain situation, untying the robot speed from the level of final *Eligibility*, while avoiding the deviation that results from a *Forbidden* direction. In absence of Brake actions the robot will take the maximum speed.

This approach is influenced by the selection of intentions introduced by [Michaud97] and the storage of information on a histogram around the robot directions is similar to the one used by Borenstein et. al.[Borenstein89]. The histograms used to represent the actions provide the basis for the proposed behaviour blending. Using this approach, convenient robot motion decisions about direction and speed will be taken.

4.3.3 Action overlapping

In order to handle all the actions, resulting from all the considered situations, the individual actions of the same kind, *Elect*, *Forbid* and *Brake*, are gathered in independent histograms that will be called *Action Maps*. Using the *Action Maps* there are no limitation about how many actions of the same kind can be taken for a single situation or how many actions can be taken for the different situations.

Three *Action maps* are then built: *Elected*, *Forbidden* and *Braked*, gathering the information of all the considered situations. Each of the *Action Maps* can be considered a space of action, providing different backgrounds or reasons to be considered in the different situations.

- The *Elected* action map gathers all the *Elect* actions, which correspond to the directions that are desirable to take, or at least acceptable, in order to handle the situation. In this way, when the *Elected action map* is built, it will represent how much desirable are each direction around the robot.
- The *Forbidden* action map will contain the information about which directions should not be taken by the robot. The final map will represent those directions that must be avoided due to the presence of an obstacle or some other problem.
- Finally the *Braked* action map will contain the speed limitations for each direction around the robot. This information will result in the actual speed to be taken when the robot moves in a specific direction.

Due to the different nature of the actions, the method followed to combine them is particular to each one.

The *Elected* action map is built by the addition of the different *Elect* actions as shown in eq. 4.20 where $E(\theta; \mu_i, \sigma_i, \delta_i, w_i)$ represents a single *Elect* action that follows the general expression introduced in eq. 4.18. Considering just the *Elect* actions, the direction that most situations state as good, will be the best movement direction for the robot. In the *Elected* action map, its highest value does not depend on a single situation contribution but in the most common contribution of all the situations, as can be seen in figure 4.13.

$$Elected(\theta) = \sum E(\theta; \mu_i, \sigma_i, \delta_i, w_i) \quad (4.20)$$

In a different way, the *Forbidden* and *Braked* action maps are built as the union of the different actions contributing to them, eq. 4.21 and eq. 4.22 were, again $F(\theta; \mu, \sigma, \delta, w)$ and $B(\theta; \mu, \sigma, \delta, w)$ represents individual *Forbid* and *Brake* actions following the general expression introduced in eq. 4.18. Here the value of the *Forbidden* action map will depend on the highest contribution of a single situation. When one situation determines a certain threat level and other situation determines a different threat level in the same direction, the robot will be safe when it reacts to the maximum threat level since this reaction will also cover the lower threat level. The same reasoning can be applied to the *Braked* action map, the most conservative speed limitation on a given direction will be the safest one to take when moving on that direction.

$$Forbidden(\theta) = \bigcup [F(\theta; \mu_1, \sigma_1, \delta_1, w_1), \dots, F(\theta; \mu_i, \sigma_i, \delta_i, w_i), \dots] \quad (4.21)$$

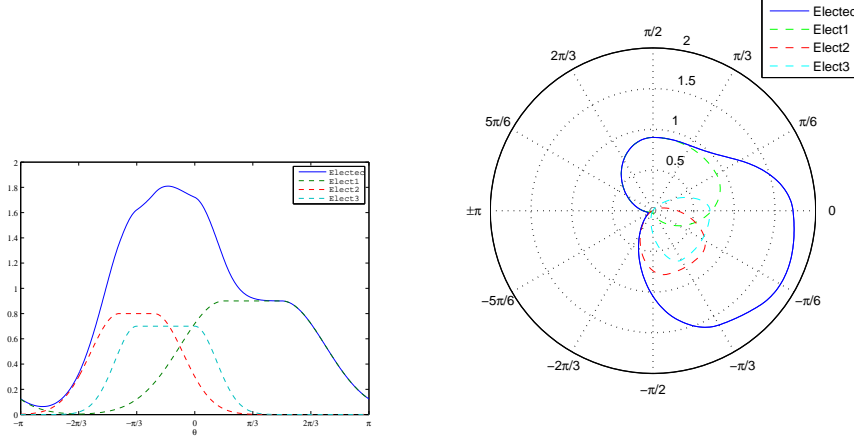


Figure 4.13: *Elected* action map built from three *Elect* actions. Cartesian and Polar representations

$$Braked(\theta) = \bigcup [B(\theta; \mu_1, \sigma_1, \delta_1, w_1), \dots, B(\theta; \mu_i, \sigma_i, \delta_i, w_i), \dots] \quad (4.22)$$

Figure 4.14 shows an example of the union of two contributions.

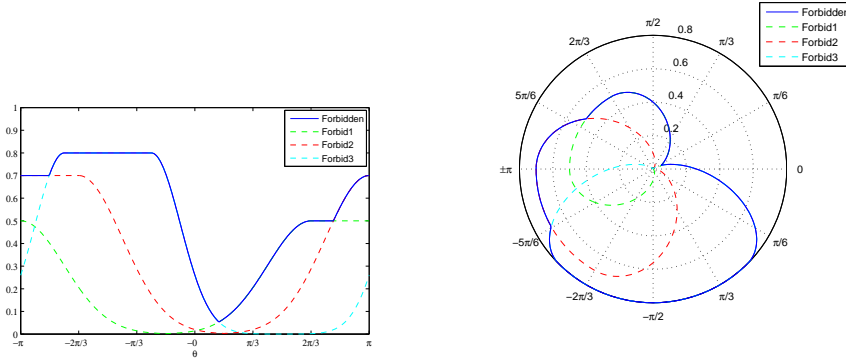


Figure 4.14: Example of the union of three contributions. Cartesian and Polar representations

4.3.4 Action Processing

Once the three *Action Maps* are built, they are used to extract the final movement direction and speed. The final movement direction is obtained from the analysis of the *Elected* and *Forbidden* action maps, while the final speed is taken as the maximum speed allowed by the *Braked* action map in the robot actual direction.

In the process to obtain the final movement direction several steps are taken:

- The *Suitability* of each direction determines the relative interest of each direction available to the robot using just the *Forbidden* and the *Elected* action maps, without considering the current facing direction of the robot.

Through the evaluation of the $Forbidden(\theta)$ and the $Elected(\theta)$ action maps, as described in eq. 4.23, a new map is built, the *Suitability* map, $S(\theta)$. In the *Suitability* map those directions that lead to a potential threat for the robot ($Forbidden = 1$) are not good directions to take ($S = 0$). Also those directions which are indifferent ($Elected = 0$ and $Forbidden = 0$) results in $S = 1$, let say a neutral suitability. Finally the direction that was elected by the situations and lead to no danger ($Elected = \max(Elected)$ and $Forbidden = 0$) results in the maximum suitability $S = 2$.

$$S(\theta) = [1 - Forbidden(\theta)] \times \left(1 + \frac{Elected(\theta)}{\max(Elected)}\right) \quad (4.23)$$

Using the sample action maps from Figures 4.13 and 4.14, a sample *suitability* map is built in Figure 4.15. There it can be seen how the *Suitability* map is similar to the *Elected* action map but where those directions that present a potential threat were take out.

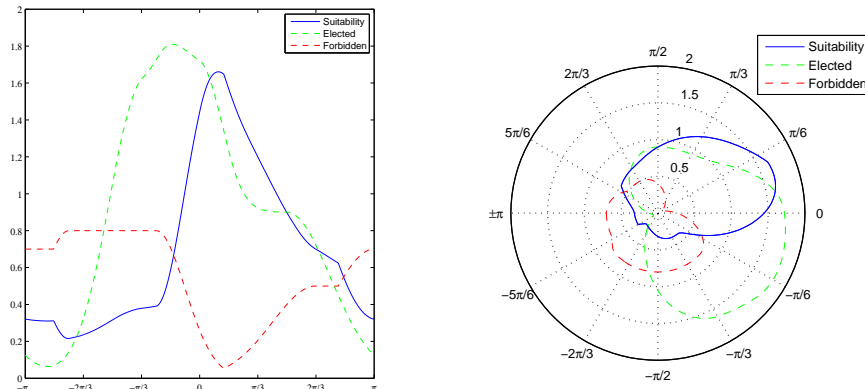


Figure 4.15: Example of *Suitability* map along with its source *Elected* and *Forbidden* maps. Cartesian and Polar representations

- Because the considered robots are non-holonomous a *Safety* map, denoted as SF is also built. This factor corresponds to the accumulated threat level involved in changing from the current direction, $\theta = 0$, to any new θ . This threat level does not only consider the threat in the final movement direction candidate, but also the threat level in all the directions that need to be crossed to reach it. Since the change of heading could be done turning clockwise, or counter-clockwise, two safety maps are computed: one for counter-clockwise turning, SF^+ , and another for clockwise turning, SF^- . The calculation of SF^+ for a given angle θ , which can be between 0 and 2π , is made according with the pseudo code found in eq. 4.24

$$\begin{aligned}
M &= 1 - \min(\textit{Forbidden}(\xi)); \quad 0 \leq \xi \leq 2\pi \\
L_0 &= 1 \\
\textit{FOR } \beta = 0 : s : 2\pi \\
\quad L_1 &= 1 - q \cdot [\textit{Forbidden}(\beta) - \min(\textit{Forbidden}(\chi))]; \quad 0 \leq \chi \leq \beta \\
\quad L_0 &= L_0 \cdot L_1 \\
\quad SF(\beta) &= L_0 \cdot \frac{1 - \min(\textit{Forbidden}(\chi))}{M}
\end{aligned} \tag{4.24}$$

The expression in eq. 4.24 is built in a way that the safety value increases while the *Forbidden* value decreases and it decreases while the *Forbidden* value is higher than the lowest *Forbidden* value already passed. To illustrate the evolution of the safety value, the safety map in figure 4.16 is built from the forbidden map shown in figure 4.14.

In eq. 4.24 $L_1 = 1$ as long as $\textit{Forbidden}(\beta)$, the currently tested forbidden value, is equal to $\min(\textit{Forbidden}(\chi))$, the lowest forbidden value already tested. When the current forbidden value is higher than the lowest tested one, the value of $L_1 < 1$.

The value of L_0 , initialized to 1, will decay when $L_1 < 1$, this decay rate will depend on the difference of the current forbidden value with the lowest tested one and on the speed factor q . The factor q , where $0 < q < 1$, states the decay rate of the safety level as a *Forbidden* space (angular) is crossed. The specific value of q depends on the granularity of the *Forbidden* map, s ; for the current work it have been set to $q = 0.1478$, which establishes a 90% reduction (from 1 to 0.1) of the value of L_0 when going through a angular distance of $\pi/6$ with a granularity $s = \pi/180$ and with difference of 0.5 between the tested forbidden value and the lower tested one.

The safety level, $SF(\beta)$, is obtained through the modulation of the L_0 value with the relation between $\min(\textit{Forbidden}(\chi))$, the lowest tested forbidden value, and $\min(\textit{Forbidden}(\xi))$, the absolute lowest forbidden value of the map.

Through this expression the safety value will only reach full safety state, $SF(\bullet) = 1$, if the lowest forbidden value on the forbidden map can be reached through a path of decreasing forbidden values.

In general the safety value at the initial direction, $\theta = 0$, will depend on the relation between the forbidden value at the initial direction and the absolute minimum forbidden value; from there, the safety will increase as the forbidden value decreases toward the absolute minimum. Otherwise the safety will decrease while the current forbidden value is higher from the last local minimum. Once the absolute minimum forbidden value is reached the safety will decay when the currently tested forbidden value differs from this absolute minimum.

The SF^- factor is computed with the same algorithm, eq. 4.24, with opportune change in the iteration direction

- One final map is built, the *Turning* map, eq. 4.25. This puts together the information obtained from the *Suitability* map with the one obtained from the *Safety* map, considering also the effort associated with the turn

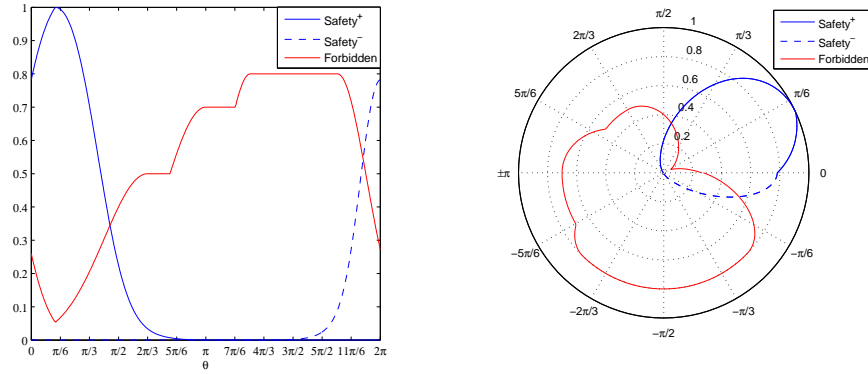


Figure 4.16: Example of *Safety* map along with its source *Forbidden* map. Cartesian and Polar representations

manoeuvre of the robot. Since there are two *Safety* maps, one for each turning direction, eq. 4.25 is computed two times, one for each *Safety* map. The cost of the turning manoeuvre is considered to increase linearly with the heading change at a constant rate w_t . The *Turning* map represents the final value, at each direction, for the turning decision

$$TM^\pm(\theta) = \left(S(\theta) - w_t \left| \frac{\theta}{2\pi} \right| \right) \cdot SF^\pm(\theta) \quad (4.25)$$

From the previous examples in figures 4.15 and 4.16 the corresponding turning map will be the one in figure 4.17

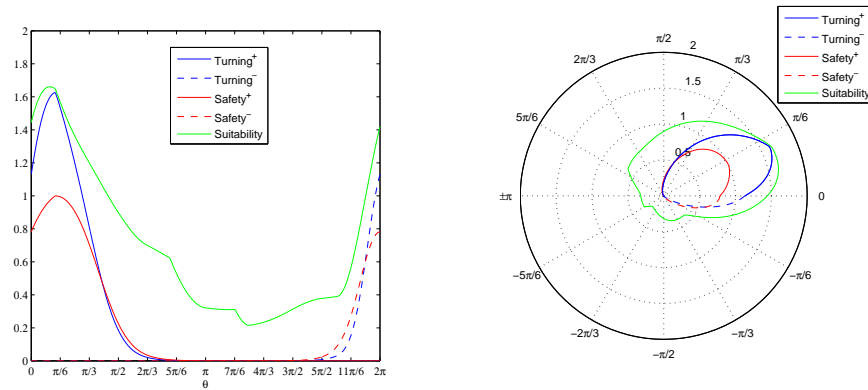


Figure 4.17: Example of *Turning* map along with its source *Suitability* and *Safety* maps. Cartesian and Polar representations

Considering the previous equations, 4.23, 4.24 and 4.25, the final movement direction to be taken by the robot will be the one with the highest $TM(\theta)$ value

associated with it, as stated in eq. 4.26.

$$\theta_{FINAL} : TM(\theta_{FINAL}) = \max(TM^{\pm}(\theta)) \quad (4.26)$$

The obtained value, θ_{FINAL} , is the reference supplied to the robot motion control loop.

Speed The other reference value for the robot motion control is the velocity used for the robot movement. This is computed for the instantaneous robot direction, $\theta = 0$, using the expression in eq. 4.27

$$\nu_{ROBOT} = \nu_{MAX} \times (1 - Braked(\theta = 0)) \quad (4.27)$$

The final values θ_{FINAL} and ν_{ROBOT} are kept as references for the robot movement control until a new evaluation of the scenario is done and a new set of maps is built. The evaluation of the scenario is done periodically with a rate depending on the robot and scenario dynamics.

Chapter 5

Studied Cases

5.1 Single Robot

The first case to study is a simplistic one, an scenario with a single robot and a single obstacle. As it has been illustrated in the previous chapter at 4.2.1, the most common problem of the classical field methods is the appearance of local minima on the field which, when using an also common gradient descent method, leads to *traps* in the field where the robot gets stuck and unable to the goal.

It is, therefore, a good exercise to use the most simple situation, in which a common approach fails, to illustrate how the proposed method works and how it is able to solve the situation.

5.1.1 System design

Used Fields Depending on the problem to be solved, the amount of needed distinct fields varies, usually along with the number of kind of elements interacting in the scenario. The most basic scenario for robot navigation have three different kinds of elements: goal, obstacles and the robot itself. As long as there is only a single robot only two distinct fields are needed: one for the goal and one for the obstacles. The expressions used to build such fields can be found in eq. 5.1 for the goal and eq. 5.2 for each obstacle. Those equations are built according with eq. 4.5 and eq. 4.6; there is no need of the use for the tuning parameter a since the two fields do not need to be merged.

$$\phi_{GOAL}(d) = d^2 \tag{5.1}$$

$$\phi_{OBS}(d) = 1/d^2 \tag{5.2}$$

Situations and Descriptors In such a simple scenario only two situations needs to be considered:

- The path to the goal is clear.
- There is an obstacle in the path to the goal.

In order to express this situations through a mathematical expression two *Descriptive Elements* are needed, one stating when an obstacle is near to the robot and another stating the presence of an obstacle in the path to the goal.

The first *descriptive element* is built using the normalization processes described along section 4.2.2, where the presence of an obstacle near the robot is described in eq. 5.3. The distance $Obs_{Distance}$ shown in eq. 5.3 and defined in 5.4 is actually the equivalent distance to the single equivalent obstacle extracted from the obstacle potential as is described in 4.16. However, since in this first scenario there is only one obstacle, this equivalent distance matches with the real distance to the obstacle.

The relative orientation of the robot to the obstacles, θ_{OBS} , is especially relevant when working with non-holonomous robots due to their manoeuvring limitations, the easiest way to deal with this limitations is to anticipate the need of the manoeuvres. The close and far distance parameters d_{Close} and d_{Far} , found in eq 5.3, are built in eqs. 5.5 and 5.6 to consider the relative orientation between the robot and the obstacle in order to improve the robot behaviour –so it does not *avoid* and obstacle at its back–; using the r_{Lim} parameter of the robot, a base distance unit that will depend on the robot characteristics is established. From the definition of d_{Far} and d_{Close} , the robot is set to take into consideration an obstacle ahead $-\theta_{OBS} = \pi-$ when it is at a distance of $6 \cdot r_{Lim}$ from the robot and it will be considered fully close at $2 \cdot r_{Lim}$. When the obstacle is at the back of the robot $-\theta_{OBS} = 0-$ these values are halved.

$$\begin{aligned} Obs_{Near} &= Sg(x; x_{001}, x_{099}) \\ x &= Obs_{Distance} \\ x_{001} &= d_{Far} \\ x_{099} &= d_{Close} \end{aligned} \tag{5.3}$$

$$Obs_{Distance} = \frac{1}{\sqrt{\phi_{Obs}}} \tag{5.4}$$

$$d_{Close} = \left(1 + \left| \frac{\theta_{OBS}}{\pi} \right| \right) \cdot r_{Lim} \tag{5.5}$$

$$d_{Far} = 3 \cdot d_{Close} \tag{5.6}$$

The specific expressions and boundary values used in eq. 5.5 and eq. 5.6 are just reasonable expressions/values set for the scenario and its objective; they are not optimal and can be modified for sharper or softer turns which provides equally good results as long as they left space enough for the robot to turn. The distance parameter d_{Close} sets the point at which the robot consider itself fully close to the obstacle –a distance where it is imperative do something to avoid the collision–. Since r_{Lim} represents the turn radius and therefore the minimum distance needed by the robot to turn $\pm\pi/2$ to avoid a frontal collision. The value of d_{Close} varies between $2 \cdot r_{Lim}$ when the obstacle is at the front –which provides space enough to avoid the frontal collisions, but not too much free margin, especially since the robot body size r_{Lim} is not explicitly considered (to simplify the expressions)– and a d_{Close} value of r_{Lim} when the obstacle is at the back, providing a minimum manoeuvring space. In the same way d_{Far} defines a *flag* distance for the presence of an obstacle. This d_{Far} distance sets the point where the robot will start to consider the obstacle for its movement,

but is not a threat yet, so it must define enough space to consider other elements and time for the robot to be ready to turn safely.

The second *Descriptive Element* needed for this scenario, $Obs_{InGoalPath}$, should state if there is an obstacle in the path to the goal. This *Descriptive Element* can be built only because it is possible to access separately to the goal field and to the obstacle field, it would not be possible to build this *Descriptive Element* in a fully unified field.

Looking at the relative orientation between the goal field gradient and the obstacle field gradient it is possible to discern if there is an obstacle in the path of the field. The *descriptive element* is built in eq. 5.7 and uses a non linear normalization to improve its performance since the obstacle –or the equivalent obstacle– is only of interest when it is clearly aligned with the goal, therefore the normalization is focused on that region.

In the construction of eq. 5.7 must be noted that the obstacle field is repulsive while the goal field is attractive so their gradients will have opposed directions if their sources were located at the same place relative to the robot. Considering this, when the two elements are aligned at the same side of the robot $\theta_{OBS} - \theta_{GOAL} \approx \pi$ and $Obs_{InGoalPath} \approx 1$ and the value of $Obs_{InGoalPath}$ will not depend on the current robot heading.

$$Obs_{InGoalPath} = \frac{1 - \cos(\theta_{OBS} - \theta_{GOAL})}{2} \quad (5.7)$$

With the mathematical expressions for the needed *Descriptive Elements* defined, the expressions for the situations to consider can be built:

- The path to the goal is clear –meaning that there are no obstacles nearby in the path to the goal–. The expression to evaluate this situation is built in eq. 5.8 by the negation of the *Descriptive Elements* defined in eq. 5.3 and eq. 5.7.

$$S_{1.0} = 1 - (Obs_{Near} \cdot Obs_{InGoalPath}) \quad (5.8)$$

- There is an obstacle in the path to the goal. Shown in eq. 5.9, the expression to evaluate this second situation is built by direct concatenation of the two *descriptive elements* defined in eq. 5.3 and eq. 5.7.

$$S_{1.1} = Obs_{Near} \cdot Obs_{InGoalPath} \quad (5.9)$$

While the expressions of $S_{1.0}$ and $S_{1.1}$ are here opposed, this is just because of the simplicity of the scenario. In general, when many situations can be possible, the complete set of situations needed to handle the scenario must cover all possible cases that the robot should face, to ensure that the robot is always in some non-zero situation. Here $S_{1.0}$ has been built as the opposite of $S_{1.1}$ as an easy way to obtain these completeness. Different expressions could have been used for both situations –not necessarily complementary– as long as they were not null at the same time .

Actions Having the expressions for the situations already defined, the next step is to establish the actions that will rule the robot movement according to the situations value.

In case that situation $S_{1.0}$ is the only active one – $S_{1.0} = 1$ and $S_{1.1} = 0$ –, the action to be taken is clear: move straight to the goal. However we should consider also that when $S_{1.0}$ is the dominant one but lower than 1, the other situation value will be greater than 0 at the same time. Therefore while the goal direction is the preferred to be taken, the sides of the goal direction can also be chosen to best adapt to those times were $S_{1.0}$ is dominant but not exclusive. Setting a preferred direction means adding a *Elect* action associated with this situation, and since there is no more considerations needed for the situation, no need of forbid any direction nor to limit the speed, the *Elect* action shown in Table 5.1 will be the only action taken for this situation.

This *Elect* action –as any other– is expressed following the expanded bell curve introduced in 4.18. For this case the central value for the action function, μ , is set to θ_{GOAL} with side the tail values around this central value, controlled by the standard deviation σ , set to $\pi/4$. Since there is only one single preferred direction for the movement the expansion term, δ , is set to zero. The weight, w , of this action –the relevance that the action will have when merged with other actions– is set to $(S_{1.0})^2$ so the action weight will remain lower –lower than the situation value– while the situation is not fully true, allowing a higher relevance of other actions –those which are linear with their situation value– to take care of robot movement when the path is only partially clear. Only in the case that the path is completely clear, this action will be fully weighted.

Table 5.1: Situation $S_{1.0}$ related actions

Distribution	μ	σ	δ	w
<i>Elect</i> (μ, σ, δ, w)	θ_{GOAL}	$\pi/4$	0	$(S_{1.0})^2$

The actions needed to deal with the second situation –when there is an obstacle in the path to the goal– are more richer than the previous one: The main action to be taken is to avoid the collision, this is done through a *Forbid* action.

Secondary to this, but somehow overlapped, is to get over the obstacle. Many approaches opt for moving around the obstacle, but actually is just necessary to move in any other direction that is not the one of the obstacle, and this is done using an *Elect* action widened in a way that cover with equally highest values all the directions that do not face the obstacle. The moving-around-obstacle behaviour will then appear as a consequence of putting together the two actions move-to-goal and move-anyway-but-to-obstacle because both have been defined as bells and not as a vectors.

A third action is also defined to deal with situation $S_{1.1}$, a *Brake* action. To impose a speed limitation while moving near an obstacle is desirable to reduce the collision danger in case of non perfect robot movement. Since the robots are considered non-holonomous, the angular space covered by the obstacle will increase during the robot avoidance movement –the limited turn radius forces the robot to move towards the obstacle in order to turn– so this must be considered in all the previously mentioned actions by widening the actions effect as

the distance to the obstacle decreases.

The set of actions associated with situation $S_{1.1}$, as defined in eq. 5.9, is shown in Table 5.2.

Having a look at the specific values of these actions we can see that the central value for the *Forbid* and *Brake* actions is center at the obstacle direction, $\theta_{OBS} + \pi$ –the repulsive nature of the obstacle field means that θ_{OBS} points opposite to the obstacle direction– while the *Elect* action is centered around the direction opposite to the obstacle θ_{OBS} . The bell expansion value, δ for the *Forbid* and *Brake* actions are proportional to the situation value, $S_{1.1}$. This will increase the top forbidden angular section as the robot approaches the obstacle.

The constant used in the *Brake* action is slightly lower than the *Forbid* one to ease the manoeuvring when the robot is moving out of that forbidden section. In a similar way the expansion term, δ of the *Elect* action is proportional to the complement of the situation value, making the *Elect* action top value section decrease as the obstacle is getting nearer.

The values set for the tail size, the standard deviation σ , follow the same criteria as the δ values widening or shortening the tails as the distance to the obstacle varies. For all these actions, the action weight is directly the situation value. By setting the weight of the actions associated with $S_{1.1}$ linear with its situation value, while the weight factor of the actions associated with $S_{1.0}$ is quadratic with it, the actions taken by the robot will be quickly dominated by the collision avoidance actions when the presence of an obstacle in the path to the goal were relevant. Here the power function applied to the situation evaluation in the action weight is used to provide precedence to one situation over the other one.

Table 5.2: Situation $S_{1.1}$ related actions

Distribution	μ	σ	δ	w
<i>Forbid</i> (μ, σ, δ, w)	$\theta_{OBS} + \pi$	$(\pi/2) \cdot S_{1.1}$	$(\pi/2) \cdot S_{1.1}$	$S_{1.1}$
<i>Elect</i> (μ, σ, δ, w)	θ_{OBS}	$\pi/4$	$\pi \cdot (1 - S_{1.1})$	$S_{1.1}$
<i>Brake</i> (μ, σ, δ, w)	$\theta_{OBS} + \pi$	$(\pi/3) \cdot S_{1.1}$	$(\pi/3) \cdot S_{1.1}$	$S_{1.1}$

5.1.2 Simulation

The case presented in 5.1.1 is tested through simulation –simulator details can be found in Appendix A– to obtain a first view on the resulting robot behaviour.

The proposed scenario is shown in Figure 5.1 where the vertical wall represents the obstacle, the flat circle represents the the goal, and the robot is represented by the irregular tetrahedron which longest side points towards the heading direction. At the right side of Figure 5.1 the obstacle and goal fields can be appreciated as they will be sensed by the robot (but the robot does not see the complete map, just the value and gradient at its current location)

Figure 5.2 shows the resulting robot path when the previous set of situations and actions are applied. The suitability map is also represented around the robot to show its evolution along the robot movement; as a reference, the thinner circle represents Suitability = 1. It is clear in the image that the robot does not get

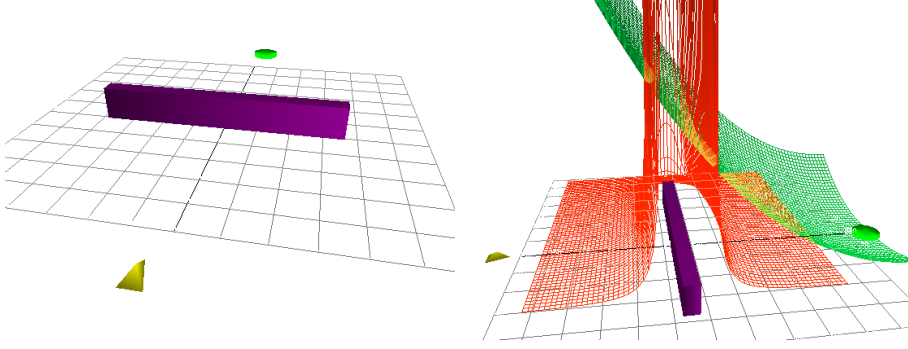


Figure 5.1: Single simulated robot scenario and visualization of the employed Fields

trapped at the local minimum trap and how the suitability decreases toward the wall when it is near to it.

Situations evolution To illustrate the proposed method, Figure 5.3 plots the evolution of the situation indicators $S_{1,0}$ and $S_{1,1}$ described in eqs. 5.8 and 5.9 along the simulation time.

- Initially the path for the robot to the goal is free of obstacles; therefore the situation described by $S_{1,0}$ have its top value since there is no obstacle in the way. On the other hand being no obstacle on the proximity $S_{1,1}$ have a minimum value; however there is an obstacle in the way to the goal, but being not in the proximity the $ObsNear$ described in eq. 5.3 collapses the final result of $S_{1,1}$.
- The second capture of the robot in Figure 5.2 corresponds to $t \approx 8s$ when the the robot approaches the obstacle and the value of $S_{1,1}$ begins to climb up while the value of $S_{1,0}$ drops down. See here in Figure 5.2 how the suitability of those directions pointing to the obstacle drop while the suitability of all other directions raise above the unit.
- In the third capture, the robot is already skirting the obstacle , this corresponds to $t \approx [10, 16]s$. During the skirting manoeuvre an equilibrium between $S_{1,0}$ and $S_{1,1}$ is reached. While $S_{1,0}$ pulls toward the goal and therefore towards the wall $S_{1,1}$ pushes for any direction that is not the wall direction. With both actions working together, when $S_{1,1}$ is higher the robot will move away from the wall so the value $S_{1,1}$ will decrease while the value of $S_{1,0}$ will rise since the obstacle is farther than previously, that will lead to a movement toward the goal and toward the wall so $S_{1,0}$ will decrease and $S_{1,1}$ will increase again returning to the initial situation but with the robot effectively moving along the wall.

Still in the third capture, it can be seen in the suitability map around the robot how those directions pointing towards the goal are under the unit value, meaning that they are not suitable while all the other directions are quite bigger than the unit. The robot will move skirting the wall not

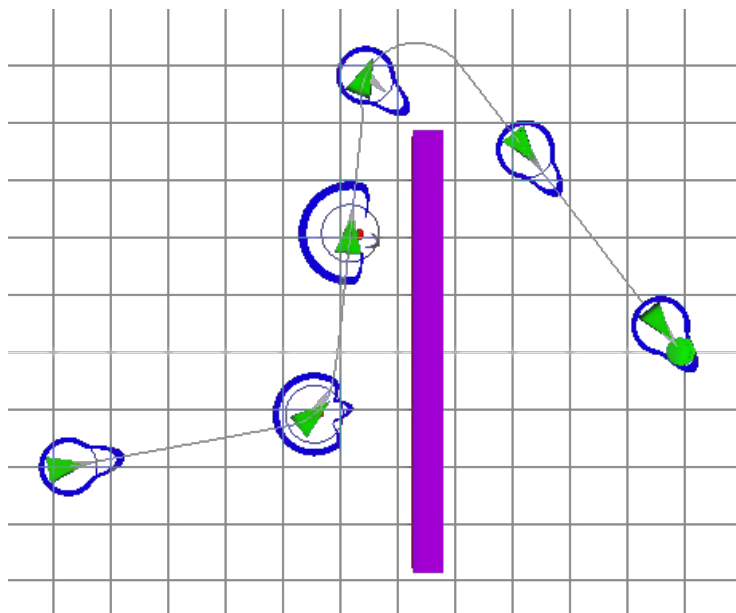


Figure 5.2: Robot path along the classic local minima trap scenario using the proposed method. Suitability map represented around the robot at each capture.

because any of the actions states the skirting manoeuvre, but because that direction is the nearest one to the robot heading suitable enough .

- In the fourth capture the robot is turning around the obstacle corner. Initially the robot is at the equilibrium situation previously described in the third capture, however, the limited turning radius of the robot prevents the robot from moving following the obstacle corner, forcing it to move slightly away from the obstacle. This results in a lapse near $t \approx [21, 24]s$ where the obstacle is no longer directly in the path to the goal, so the value for $S_{1,0}$ raises while the value for $S_{1,1}$ decreases.
- Between the fourth and the fifth capture, as the robot moves towards the goal, it approaches the obstacle again. During this period the distance between robot and wall reaches lesser values than the distance reached in the second or third capture. This happens because the relative location of the obstacle from the robot is different, therefore $S_{1,1}$ does not take a higher value in the peak of that period, near $t \approx 26s$ in Figure 5.3. The result is that there is little disturbance affecting the robot path, even when the robot is at the nearest point to the obstacle.
- Finally, the last capture shows the robot again clear from the obstacle and free to move towards the goal. The values of $S_{1,0}$ and $S_{1,1}$ were again those of the beginning, where $S_{1,0}$ fully controls the situation.

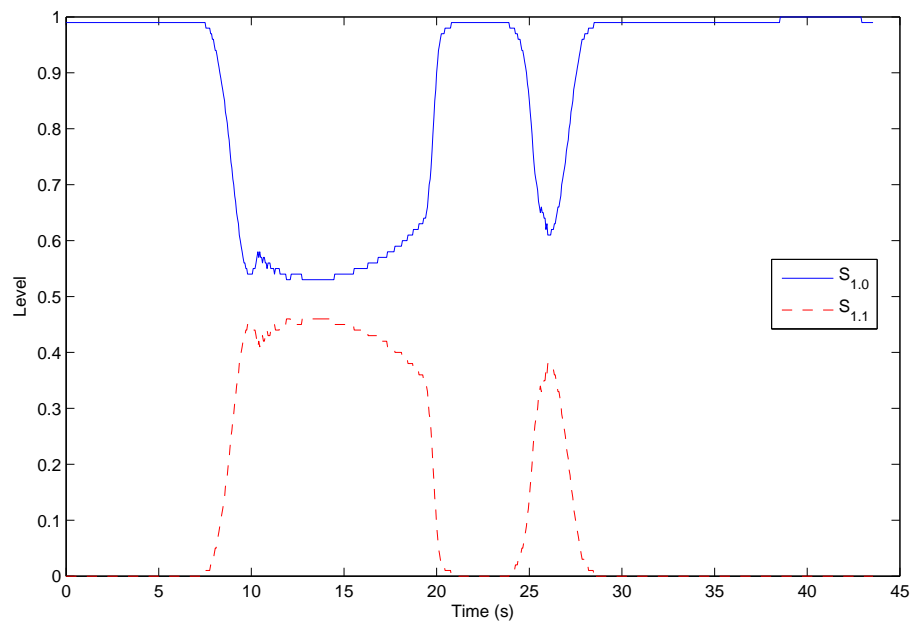


Figure 5.3: Evolution of situation indicators $S_{1,0}$ (solid) and $S_{1,1}$ (dash) computed by the robot along the simulation time

5.2 Multiple, independent robots

The procedure used to build the navigation scheme of a single robot in search for a goal can be expanded for the navigation of multiple robots in the same scenario. In this scenario the robots will behave as independent entities in the search for a goal, however they will need to take care of the presence of other robots, avoiding now collisions with both obstacles and other robots.

5.2.1 System Design

In order for the robots to move to the goal in the presence of obstacles and other robots, each individual robot will behave basically like in the single robot example, however new behaviours should to be introduced to avoid the robot-robot collisions.

Robot field The robot-robot collisions cannot be handled by the previously used potential fields defined in eq. 5.1 and eq. 5.2 because there is no information in the system related to the other robots. A new field is then needed to cover this robot-robot interaction. The new potential field is targeted to avoid collisions so it will be repulsive, and since the different potential fields are not merged it will have the same expression as the obstacle field in eq. 5.2. However this new field will be sourced by each robot in the scenario and only merged with the fields of other robots, being kept apart from the obstacle and the goal fields. The potential associated with each robot is expressed in eq. 5.10, where d represents the robot to robot distance. In the potential and field sensing process only the contributions being sourced by other robots are considered, the field being sourced by the sensing robot is ignored.

$$\phi_{ROB}(d) = 1/d^2 \quad (5.10)$$

Descriptive elements and Situations

Obstacles The descriptive elements related with the obstacles used for the case of the single robot scenario can still be used in this situation. So in this multiple robot scenario the descriptor for obstacle proximity, Obs_{Near} , defined in eq. 5.3 and the descriptor for the presence of an obstacle in the path to the goal, $Obs_{InGoalPath}$, defined in eq. 5.7 will be used as they are, no modification are needed.

Other robots In addition to the previously defined descriptors, some new ones are needed to characterize the relation of the sensing robot with the other robots in the scenario. In a similar way to the descriptive elements defined for obstacle proximity, the proximity of other robots is built by a descriptor equivalent to the one in eq. 5.3. The new descriptor is defined in eq. 5.11 and the two reference distances are defined, again, relative to the robot manoeuvring capabilities set by r_{Lim} . Now, the relative orientation of the sensing robot to the other robot position is not considered since the other robot is also moving, so it should be fully considered in any case –leaving a robot at your back will not ensure that a collision will not happen–

$$\begin{aligned}
Rob_{Near} &= Sg(x; x_{001}, x_{099}) \\
x &= 1/\sqrt{\|\Phi_{Rob}\|} \\
x_{001} &= 1 \cdot r_{Lim} \\
x_{099} &= 3 \cdot r_{Lim}
\end{aligned} \tag{5.11}$$

Along with the the new robot distance descriptive elements, some descriptive elements have to be defined to establish the relative positions of the robots. As in the *ObsInGoalPath* case of eq. 5.7, this is done thought the relationship of the different gradient directions. Several descriptors are defined, in Table 5.3, to allow the description of specific behaviours for different situations. These descriptors are illustrated in Figure 5.4 showing their value for the different sensed orientations of the robot field relative to the sensing robot orientation.

Table 5.3: Descriptive elements related with the sensed robot field orientation

Robot sensed ahead	$Rob_{Ahead} = \frac{1 - \cos(\theta_{ROB})}{2}$
Robot sensed at the back	$Rob_{Back} = \frac{1 + \cos(\theta_{ROB})}{2}$
Robot sensed at a side(any)	$Rob_{Side} = \sin(\theta_{ROB})^2$

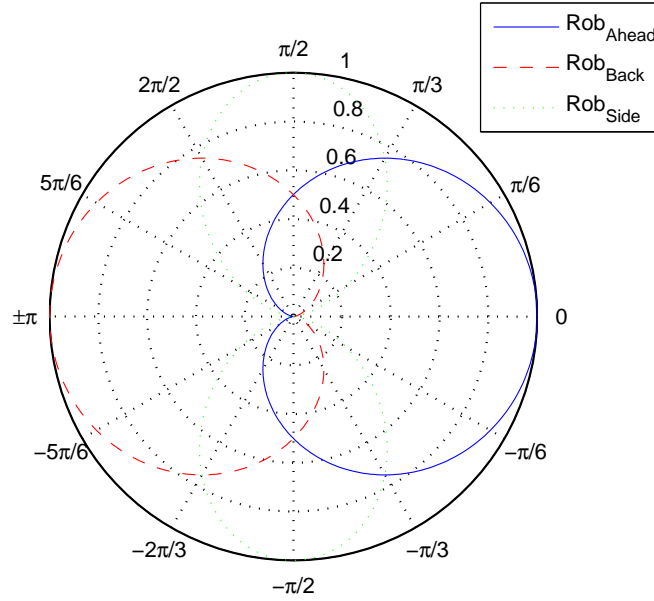


Figure 5.4: Descriptive relative positions between the sensing robot and the equivalent robot in polar representation

When there are many robots in the scenario, the measurement of the robot-robot used along the analysis in eq. 5.11 no longer represents the real distance between our robot to some other individual robot but it represents the distance to a equivalent robot in the same way that the equivalent obstacle was previously introduced. Having all the robots present in the scenario acting as source to the robot potential field, the final Φ_{ROB} is the addition of all these contributions. The distance value employed by the sensing robot for its situation analysis is extracted of this final field in the same way as it is described for the obstacles in eqs. 4.15 to 4.17.

This situation illustrates one of the advantages and one of the disadvantages of working with virtual fields mentioned at the firsts chapters. The advantage is that the information from many robots, many sources, is merged together in a set of easier to use values that will not depend on the amount of robots; using this fixed number of values, instead of the information of all individual robots, for the behaviour analysis results in easier and lighter computation. The disadvantage is the lost of the detailed information about each robot, that will provide better comprehension of the situation at a cost of processing much more information, growing linearly with the number of robots.

Figure 5.5 illustrates the concept of the equivalent robot; The final robot field resulting from the addition of the individual fields sourced by two robots is seen by the sensing robot as a third robot that is not in the position and distance of any of the two real robots.

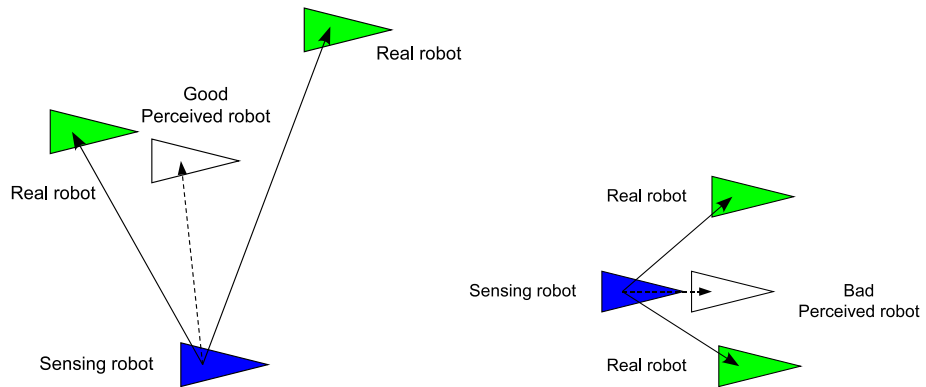


Figure 5.5: Two robots in the proximity of the sensing one in two configurations showing the perceived robot in a good and bad representation of the real ones

One danger of this merging method can be found in the classic potential field approaches which uses some kind of gradient descent algorithm to guide the robot: in some situations, around saddle points, the resulting gradient is not enough to represent the actual situation and can be confusing for the robot analysis of the situation –especially for non-holonomic robots–. Those situations can be easily identified through the relationship between the potential value and the field module, so a new descriptive element is introduced to identify those especial situations. The expression in eq. 5.12 just compares the result of the measured field module, $|\vec{\nabla}\phi_{Rob}|$ with the value of the field module that will be

sourced by a single robot in a position of equal potential, $2\Phi_{Rob}^{3/2}$. When this quotient is high, ≈ 1 , the resulting values for the equivalent robot can be trusted because it represents the most significant robots are far enough or not around the sensing robot, so there is not a saddle point in the vicinity of it. The opposite situation, where the quotient is low ≈ 0 , means that the most significant robots are all around and near the measuring robot, which means that equivalent robot is not a good representative on the real robots positions and that the sensing robot is near to a saddle point of the robot potential.

To build the descriptive element, the transition of the possible values of the quotient is handled by a sigmoid function. The transition limit values has been roughly established after obtaining the quotient result for several robot configurations.

$$\begin{aligned}
 Rob_{SaddlePoint} &= Sg(x; x_{001}, x_{099}) \\
 x &= \frac{|\vec{\nabla}\phi_{Rob}|}{2|\Phi_{Rob}|^{3/2}} \\
 x_{001} &= \frac{1}{2} \\
 x_{099} &= \frac{1}{3}
 \end{aligned} \tag{5.12}$$

Situation specification and actions The addition of other robots to the scenario, and the possibility to collide with them, makes necessary to consider them in the set of defined behaviours, however this modification will not affect all the situations.

Clear path With the new robots present in the scenario, the situation named as $S_{1,0}$ in eq. 5.8 for the single robot scenario that evaluates when the path to the goal is clear needs to be modified for this scenario because now 'clear' must consider also that there are no robots. The new situation for this new scenario can be found in equation 5.13. Now this situation evaluate that there is no obstacle near and in the path to the goal and that there is no robot –the equivalent one– near and ahead of the sensing robot; if any of those two sets of conditions is not meet the situation evaluation will fall to its lowest value. While the expression of the situation is modified the associated actions are not, so the actions specified in Table 5.1 are still valid and will be used as they are.

$$S_{1,0} = (1 - Obs_{Near} \cdot Obs_{InGoalPath}) \cdot (1 - Rob_{Near} \cdot Rob_{Ahead}) \tag{5.13}$$

Obstacle collision avoiding On the other hand, the situation used in the single robot scenario to evaluate if there are obstacles in the path to the goal, expressed in eq. 5.9, remains valid and also the associated actions to avoid the collision located in Table 5.2.

Robot collision avoiding While obstacle avoiding and robot avoiding are mostly the same, the main difference between an scenario with robots and an scenario with obstacles is that the obstacles are considered static or with known movement patterns while the other robots in the scenario are moving and active elements. All the robots in the scenario are reactive meaning that while one robot is avoiding a second one is highly possible that the second robot,

at the same time, is trying to avoid the first one. In general, the strategy used to avoid obstacle collisions, to move away from it, is also valid to avoid robot collisions, but with robots their relative positions must be also considered to establish how the other robot avoiding manoeuvre can affect the result. The relative position of the other robot is now important because we must assume that the other robot will also be avoiding us, so there is a need to avoid actions leading to manoeuvring jams like the two opposing robots moving to the same side to avoid each other resulting in a new collision or two parallel robots trying to cross each other paths moving along the side of the other one at the same speed and direction, which will result in never being able to pass along the other robot.

To avoid these robot-robot collisions two new situations are considered for the multiple robot scenario: When another robot is approaching from behind and when another robot is getting closer ahead.

- The situation expressed in (eq. 5.14) evaluates when other robots – actually the equivalent robot – are sensed near and ahead of the sensing robot. The two first terms, Rob_{Near} and Rob_{Ahead} perform this evaluation and the third term, $(1 - Rob_{SaddlePoint})$ evaluates when the two first ones really represents the direction of the actual robots.

The saddle point could be also considered in the obstacle proximity evaluation, however if the robot is moving towards a obstacle field saddle point it also implies that the robot is moving towards a passage between obstacles quite narrow, so the safest action will be to act equally as if it was a solid obstacle.

$$S_{2.1} = Rob_{Near} \cdot Rob_{Ahead} \cdot (1 - Rob_{SaddlePoint}) \quad (5.14)$$

The actions associated to this situation can be found in Table 5.4, these actions are equivalent to the evading actions taken for the obstacles found in Table 5.2. However there is a main difference between the two situations and their actions. While in $S_{1.1}$ the actions are taken if the obstacle – the perceived one – is in the path of the goal, no matter on the relative position of the obstacle to the robot, in $S_{2.1}$ the actions are taken when the other robot is ahead of the robot, no matter if it is or not in the goal direction, because the other robot can be moving toward the sensing one. This also means that a robot facing the side or the back of other robot will perform the evading manoeuvre while the other robot will not, only in the case of the two robots moving face to face the two robots will perform the evading manoeuvre.

Table 5.4: Actions related with situation $S_{2.1}$

Distribution	μ	σ	δ	w
$Forbid(\mu, \sigma, \delta, w)$	$\theta_{ROB} + \pi$	$(\pi/2) \cdot S_{2.1}$	$(\pi/3) \cdot S_{2.1}$	$S_{2.1}$
$Elect(\mu, \sigma, \delta, w)$	θ_{ROB}	$\pi/4$	$(3\pi/4) \cdot (1 - Rob_{Near})$	$S_{2.1}$
$Brake(\mu, \sigma, \delta, w)$	$\theta_{ROB} + \pi$	$(\pi/3) \cdot S_{2.1}$	$(\pi/6) \cdot S_{2.1}$	$S_{2.1}$

- The second situation related to the robot-robot collision avoidance, expressed in eq. 5.15 cover those other cases when the other robots are not strictly near and ahead and its associated actions have been set to easy the general group movement. In similar way to those found in normal car traffic, when another car approaches from the back with the intention of passing you, you move yourself slightly aside to let the other car pass. The actions for these behaviour can be found in Table 5.5.

Observing the expression of eq. 5.15, the power functions can be seen affecting both terms. Since the power factor is less than the unit this means that the situation value will quickly rise –quicker than their descriptive elements value, similar to that seen in figure 4.5– so the associated action will be promptly taken if not forbidden by any other factor. The power factor on the terms of eq. 5.15 means that the value of $S_{2.2}$ will rise even when the sensed robot is not directly at the back and when the sensed robot comes slightly near to the sensing robot. That the action is promptly taken does not mean that the action itself is harsh. The value of μ set for the central value of the Elect action at Table 5.5 – the desired movement direction –, is just a third of the approaching angle of the other robot. Therefore the sensing robot will deviate just slightly to ease the other robot avoiding manoeuvre. It will be the other robot, probably working under $S_{2.1}$, which will do the most of the avoidance.

$$S_{2.2} = Rob_{Near}^{1/2} \cdot Rob_{Back}^{1/2} \quad (5.15)$$

Table 5.5: Actions related with situation $S_{2.2}$

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	$\theta_{ROB}/3$	$\pi/4$	0	$S_{2.2}$

5.2.2 Simulation

Again, in order to study the designed system, tests have been performed through simulation. An scenario with four independent robots aiming for two different goals and one obstacle in the middle have been built and run. The shown scenario has been selected because it forces robot-robot interaction, collision situations, and also the proposed behaviour/action blending mechanism is stressed by the resulting situations where both robot-robot and robot-obstacle are present.

The initial situation of the scenario is shown in figure 5.6, here the four robots, represented as triangles, the goal, circles, and the obstacle, rectangle can be seen. The two robots on the right aim for the goal at the left side and the two robots in the left have the right side goal assigned.

The initial positions of the robots have been tuned to observe the different interaction between the robots:

- The two robots at the left side, being in line and aligned with the goal, should initially move in a queue keeping a safety distance.
- The two robots at the right, being in parallel, will initially try to converge to the same path but the collision avoidance will work to prevent this.

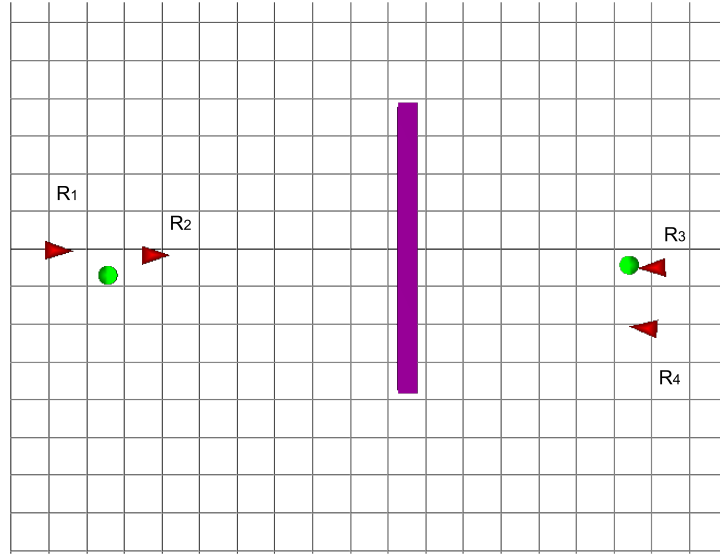


Figure 5.6: Multiple independent robots, capture 1. Initial positions of the robots, goals and obstacle for the multiple robots example.

- Being the goals at the different sides of the scenario the two group of robots will move one against the other so the case when two robot moves front to front will appear.
- Finally the obstacle position will enhance robot-robot encounters with different relative positions between robots. The obstacle will also stress the action blending method by forcing the robots to situations were they must 'choose' to move towards an obstacle or to move towards another robot.

Along figures 5.7 to 5.11 we can see the evolution of the scenario. At the end the two sets of robots arrives their goals but, as can be seen along the different captures, multiple interactions happens along the paths. Also in figures 5.12 and 5.13 the situations and speed evolution for one of the robots is shown to illustrate the the simulation analysis. This shown robot, 'Robot 3', is the top right one at the first frame.

In figure 5.7 we can see that the initial movement of the robots is straight towards their assigned goals but keeping a safe distance from their goal mate. It can be observed how robot 1 have increased the initial distance with robot 2 along this first movement. Also it can be seen how robot 4 in its movement towards the goal also moves towards robot 3, but slightly before turning due to the presence of the obstacle, robot 4 have move in parallel with robot 3. In figure 5.12 this period goes between $t = 0s$ and $t \approx 8s$ and it can be seen how the situation descriptors for the robot presence, $S_{2,1}$ and $s_{2,2}$ grows. Neither of them grow to much because robot 4 senses robot 3 at its side –actually the equivalent

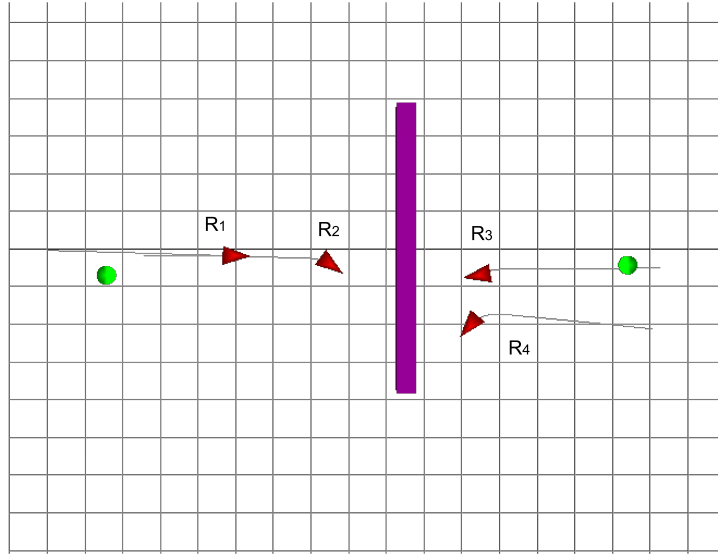


Figure 5.7: Multiple independent robots, capture 2.

robot, which is dominated by robot 3 since it is clearly the nearest one— and the presence of robot 4 is also being sensed —and considered for avoidance— by robot 3.

In the third capture of the simulation, Figure 5.8, the robots have met the obstacle and turn to avoid the collision with it. But, while in this frame the obstacle situation is already solved some new interactions have taken place. Robot 1, being behind robot 2, need to deal with no other thing than the obstacle avoidance. In the case of robot 3 and robot 4, they were mainly parallel when they have confronted the obstacle at figure 5.7 and due to their relative positions to the goal both of them have to turn to the same side to avoid the obstacle. Therefore robot 3 is in a situation where it must decide to move towards the obstacle (forbidden) or to move towards another robot that it is too close (also forbidden) —a third, common, option is to skirt the obstacle along the other side, but the initial positions of the robots have been adjusted to reduce the weight for this solution and force the current dilemma—. At the Robot 3 situations, figure 5.12, this period goes from $t \approx 8s$ to $t \approx 15s$. Along this period it can be seen the dramatic ascension of $S_{1.1}$, the obstacle collision indicator, and also how the robot proximity indicators $S_{2.1}$ and $S_{2.2}$ rise. Around $t \approx 10s$ the robot proximity and obstacle proximity are both near 0.5, where in the single robot simulation the stability between situations is achieved, but in this case that stability is not achieved. Not being able to turn to one or the other side the action of robot 3 is to reduce its speed, this can be appreciated in figure 5.13, and that speed reduction is kept until the values of $S_{2.1}$ and $S_{2.2}$ fall when the robot is free to turn, slowly because it is near an obstacle, in the direction of robot 4. Once this 'dilemma' is solved —become less relevant— for robot 3, the behaviour moves to the easier task of moving toward the goal while avoid colliding with the obstacle near $t \approx 15s$

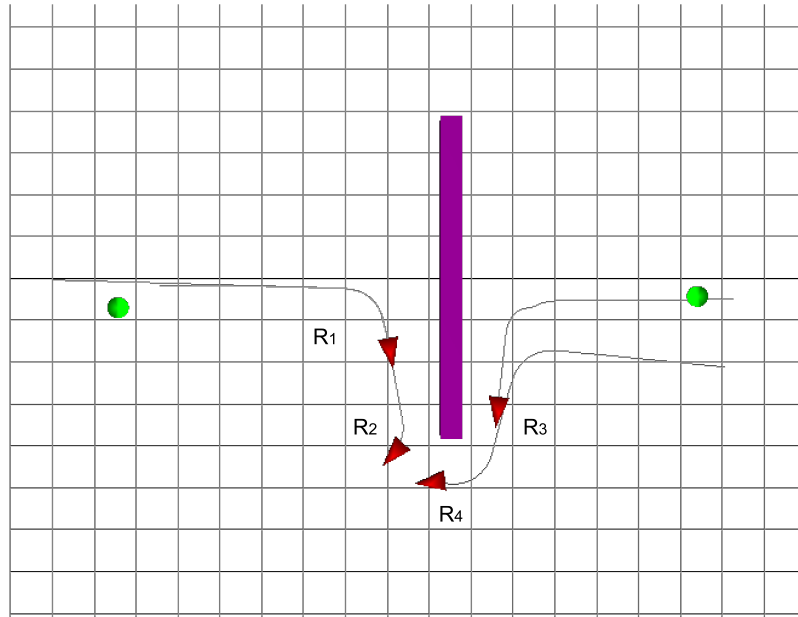


Figure 5.8: Multiple independent robots, capture 3.

In the meanwhile, at the last moments of this period, we can see how robot 4 reaches the bottom of the obstacle a little bit earlier than robot 2, so when robot 2 is going to move around the corner of the obstacle it finds robot 4 just in front. The result of this is robot 2 turning to avoid robot 4; $S_{2,1}$ is dominant there for robot 2 because robot 4 is at the front, while robot 4 is not modifying its direction yet because for him, robot 2 is approaching from the side so robot 4 has preference according to the used set of rules. If the robot-robot collision were treated as the robot-obstacle collision both robots 2 and 4 will have turn to avoid the other one by the same amount, and that will have ended with both robots side to side and moving parallel indefinitely.

Along the period ending at the fourth capture, $t \approx 15$ to $t \approx 23$, figure 5.9, we can see how robot 4 has finally manoeuvred partially to avoid collision with robot 2—they are non-holonomous, they need space to turn— but the most of the avoiding manoeuvre have been performed by robot 2, which having robot 4 at the front and robot 1 coming from behind, has ended turning completely around. Robot 1 has also moved slightly to avoid collision with robot 2, but it was surrounded by all the other robots and the obstacle. For robot 3 we can see that it has recovered its full speed and has follow the route of robot 4 around the end of the obstacle. At the time of the fourth capture robot 3 have just find robot 1 in front of it ant the end of the obstacle.

The last troublesome moment of the simulation comes between $t \approx 23$ to $t \approx 30$, shown in the fifth capture, figure 5.10. At the period between the fourth and the fifth capture, around $t \approx 24$, we can see in 5.12 how the clear path indicator, $S_{1,0}$, drops while the robot collision at front, $S_{2,1}$, peaks. This

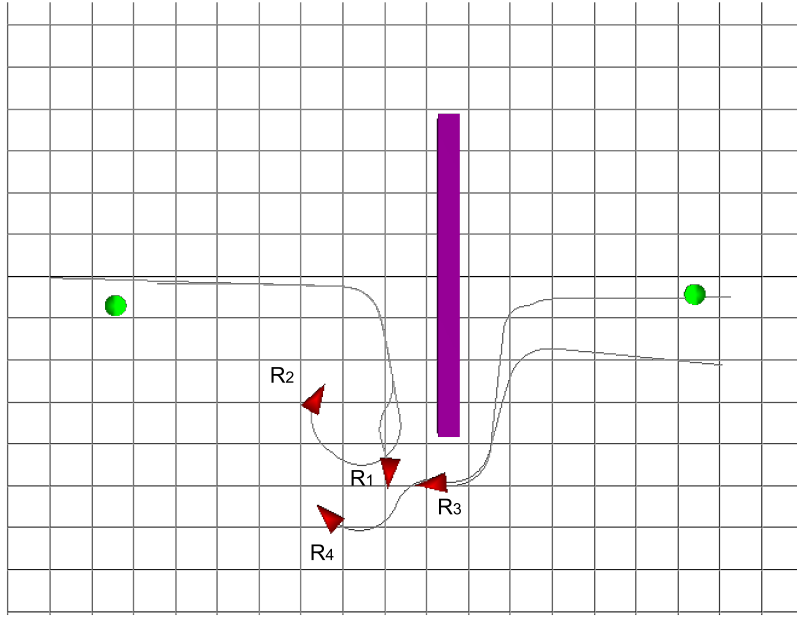


Figure 5.9: Multiple independent robots, capture 4.

is the moment when robot 3 encounters robot 1 at the corner of the obstacle. At that point turning to the right is not an option since the obstacle is there, the straight direction is occupied by the sensed robot –robot 1 is dominant but robot 4 is close enough to affect the equivalent robot– and there is no space to turn to the left. The situation is solved again by reducing drastically the speed as can see in figure 5.13. This behaviour is kept until the collection of treats reduces itself enough to allow a safe movement for robot 3. At that same point robot 1 is less threatened, robot 3 is at the side not in front, so robot 1 is able to move out and release robot 3 from the dead-end situation.

The non-holonomous nature of the robots force them to move forward in order to turn sideways, so when avoiding an obstacle directly ahead to the robot, the avoidance manoeuvre moves the robot even closer to the threatening direction. This is the main reason for the construction of the safety map in eq. 4.24, if the robots were holonomous this map would not be necessary.

Finally at Figure 5.11 we can see that robots 3 and 4 have reached their goal and robots 1 and 2 are going to do the same now that their paths are clear.

The scenario is successfully completed by the robots without collisions with the obstacle or with other robots while preserving the safe distances all the time.

The proposed methodology is a reactive local navigation system, were each robot only cares about the other robots when they are within its sensing range. This method is not optimal but it can adapt instantly to new information. Even if each robot applies –individually– a high level path planning algorithm based on local information, this algorithm will be forced to recalculate the path to

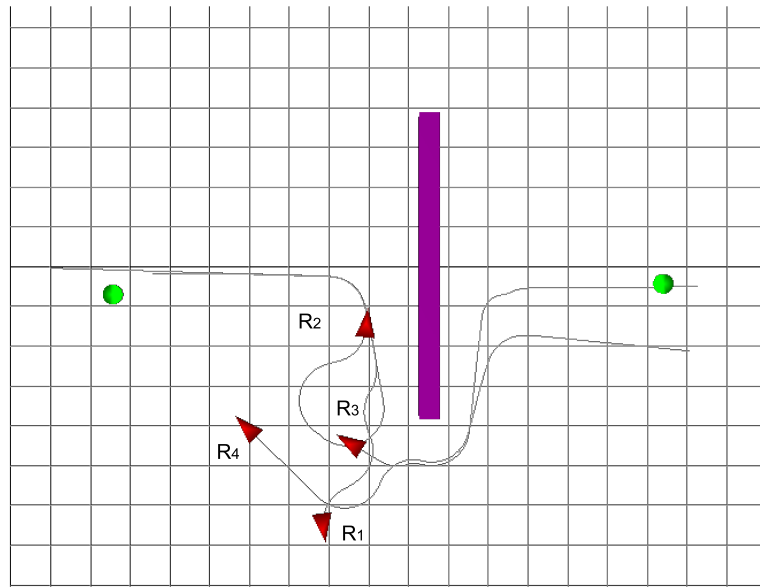


Figure 5.10: Multiple independent robots, capture 5.

the goal once and again –which is generally costly–, because the presence of the other robots was not constantly perceived. The optimal path can only be achieved having a full knowledge of the scenario at the very beginning.

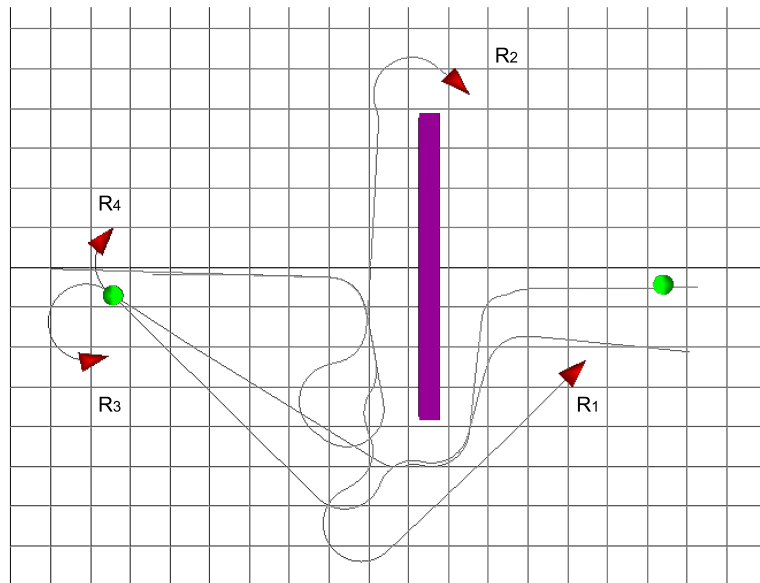


Figure 5.11: Multiple independent robots, capture 6.

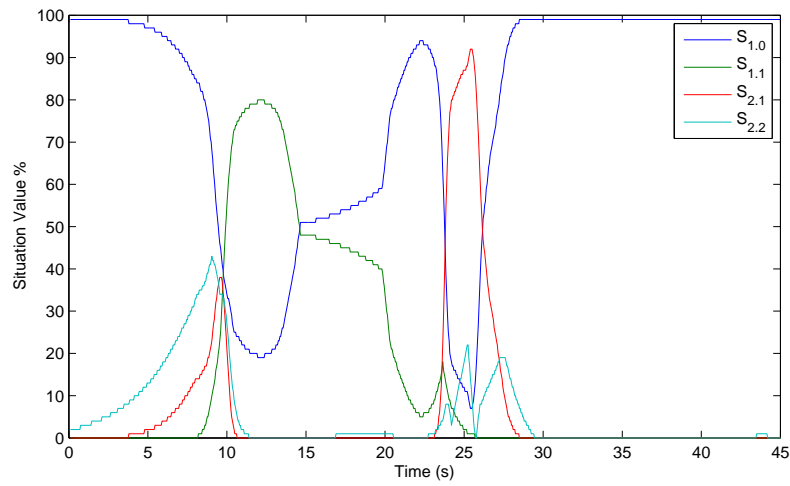


Figure 5.12: Robot 3 Situations.

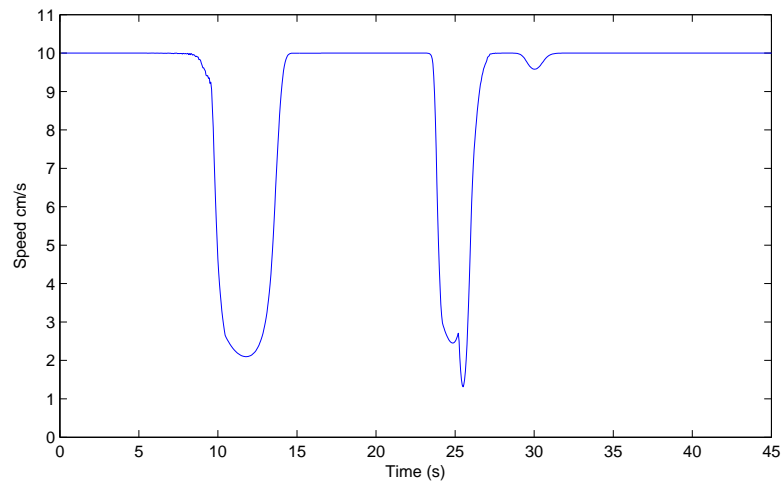


Figure 5.13: Robot 3 Speed.

5.3 Unorganized Robots Group

To provide some organization scheme is the next natural step once multiple robots, able to navigate the same space, are available. In the previous sections the robots had moved around independently, there may be other robots in the vicinity, but the other robots were simply things that gets into the way, each robot was an independent element which only cares about reaching its goal. A simple scheme where an undetermined number of robots are able to behave like a group is now studied. However this group is not designed to show any kind of explicit structure or hierarchy, just to stay together. The only requisite for the group to behave in a reasonable way and be controllable is that the goal must be common to all robots in the group.

5.3.1 System Design

While the new characteristic, belong to a group, has to be introduced, most of the previously defined elements for obstacle and robot collision avoidance can be reused for this scheme. The existence of a new entity on the scenario, the group, implies the addition of new, distinctive, set of information to the environment.

Team Field Along this work, the environment information is handled through the use of virtual fields, therefore, a new field is now introduced to hold the group information. This new field, team field, is being sourced by each individual belonging to the group, using the superposition properties of the virtual fields the general team field is built.

To build up a team, the information referring to the individuals far from the group is more relevant than the information about the individuals that are already gathered.

In the construction of the equivalent obstacle and equivalent robot, out of the composed obstacle and robot fields, it have been shown that in the repulsive fields the nearest field sources were the dominant ones.

In the case of the attractive fields, when multiple sources are added to build a single field, the dominant elements are those farther from the sensing point. Therefore when the single equivalent source is built, it will be located closer to the farther sources than to the nearer ones.

The team field is built as an aggregation of attractive field sources attached to each robot on the team, with each source following the expression found in equation 5.16 and the final field built as in eq. 5.17.

In eq. 5.16 the parameter d represents the distance between the source and the measuring point. This distance is again the robot to robot distance already employed in the construction of the robot field, no new information has to be shared among the robots, but it is now used in a different way for the construction of the team field than it was used for the construction of the robot field.

$$\phi_{Team}(d) = d^2 \quad (5.16)$$

$$\Phi_{Team} = \sum \phi_{Team}(d_i) \quad (5.17)$$

Descriptive elements and Situations

Goal The same goal field, Φ_{GOAL} , described in equation 5.1 is employed for the group goal along the study of this case. However in this case the individual robots do not aim directly for the goal, as in the previous cases, meaning that the situation $S_{1.0}$ described in eq. 5.8 and its associated actions found in table 5.1 are no longer considered for the building of this case.

Obstacles The situation and descriptive elements used in the previous examples to deal with the obstacles are used again in the building of this system. The descriptive element about obstacle proximity, $Obs_{InGoalPath}$, previously defined in equation 5.7, is still necessary, however, since the robots do not longer have their movement always intended towards the goal –they should also take care about the team–. The new descriptive element referring to obstacle relative position is defined in equation 5.18 and now its value raises whenever an obstacle is ahead of the robot, in the direction of the goal or any other one. Consequently the collision warning, handled by situation $S_{1.1}$, needs to be defined again and is expressed in equation 5.19. The actions associated to $S_{1.1}$ are still valid and applied without modification as they were defined in table 5.2

$$Obs_{InPath} = \frac{1 - \cos(\theta_{Obs})}{2} \quad (5.18)$$

$$S_{1.1} = Obs_{Near} \cdot Obs_{InPath} \quad (5.19)$$

Other Robots The robot field, introduced in equation 5.10, is built in the same way that was built in the Multiple Robot case however not all the situations described for that other case are, in this new case, applicable. The main difference with the behaviour between robots in these two cases resides in the nature of the relationship between the robots. In the multiple independent robot case, the behaviour of each robot was designed with the knowledge that the other robots present at the scenario can or cannot share the goal of the sensing robot, therefore all must be avoided the same way. In the current case, unorganized robots group, the behaviour is designed in the assumption that the other robots present in the scenario belongs to the same group as the sensing robot, so there is no need to avoid them at all times.

When moving along a group there is no need to actually avoid other group member since all members share your same goal so many times all the robots will just move in parallel. Some kind of collision avoidance is compulsory, however to reduce the speed in the direction of other group member will be usually enough, and just in some cases there is need to avoid other member direction.

The situation $S_{2.1}$ previously defined in eq. 5.14, intended to avoid the robot-robot collisions, is used again, but not all the actions associated to it will be taken for this case. Only the restrictive actions, *Forbid* and *Brake*, associated with $S_{2.1}$ will be used, leaving only the reduced set of actions defined in table 5.6.

For this same reason the situation $S_{2.2}$ found in equation 5.15 and its associated action found in table 5.5, which guide the robot to actively move around the other robots in order to reach the goal, is not considered along this case because there is no need to actively move around other team member – in a

face to face collision the forbid action on table 5.6 could result in a robot moving around another robot to avoid collision, but that is a collision avoidance mechanism, not actively moving around a team partner to reach the goal–.

When multiple groups share the same scenario, they must have specific team fields for each group and the active avoidance established to manoeuvre around non team members. The inclusion of the active avoidance behaviour described in $S_{2,2}$, when applied among members of the same group will result in group instabilities, where each robot at the back of the team will tries to move around the ones at the front.

Table 5.6: Actions related with situation $S_{2,1}$ for Robot Groups

Distribution	μ	σ	δ	w
<i>Forbid</i> (μ, σ, δ, w)	$\theta_{ROB} + \pi$	$(\pi/2) \cdot S_{2,1}$	$(\pi/3) \cdot S_{2,1}$	$S_{2,1}$
<i>Brake</i> (μ, σ, δ, w)	$\theta_{ROB} + \pi$	$(\pi/3) \cdot S_{2,1}$	$(\pi/6) \cdot S_{2,1}$	$S_{2,1}$

Team The gathering of the robots and the movement of the group towards the goal are the two main objectives of this case. To go on with these needs, a new set of descriptive elements and situations, mainly associated with the team field, are here introduced. There are several different ways to establish actions that fulfil the objectives for the group formation and movement, the one described here is not unique, nor the best one, it is used mainly because of its simplicity.

In order to accomplish the scenario objective two main situations can be established for this case:

- The robot considers itself connected to the group, so the main movement intention will be to get closer to the goal.
- The robot considers itself disconnected from the group, so the movement will be oriented to join the group.

The definition of these two behavioural groups needs the establishment of some indicator about the distance between the sensing robot and the 'group'. This descriptive element establishes when the group is near, so the sensing robot belongs to the group core, or when the group is far, so the sensing robot needs to move towards the group.

While term 'group' is loosely used here as an entity, actually is just an abstract figure, there is not a single element or entity that can be considered 'the group' and can be used to measure the distance to it. The information about 'the group' is held by the team field, which is built out of the field sources attached to each member of the group. The measurement of the team field is the one that provides the sensing robot its position among the group.

A simple way to establish when a robot is within the group is to establish a reference distance from the sensing robot to the equivalent team source –built in the same way as the equivalent obstacle and the equivalent robot–. This reference distance is established through the field that would be sensed by the robots at the sides of the group when all the robots in the team are placed in line within a given distance. Placing the n robots belonging to the group in a straight line with a distance between robots e , the value for the field module at

the side robots will be the one in eq. 5.20 and the distance to the equivalent team source using this field module will be the one stated in eq. 5.21.

$$\|\vec{\nabla}Team_{Line}\| = \sum_{i=1..n_{Sources}} 2 \cdot i \cdot e \approx n_{Sources}^2 \cdot e \quad (5.20)$$

$$d_{TeamLim} = \frac{1}{2} \|\vec{\nabla}Team_{Line}\| \quad (5.21)$$

The descriptive element for the team distance is then built in eq. 5.22 using a sigmoid function with the far limit, eq. 5.24, placed at the approximate field value sensed by a robot at the side of a group arranged as a line with r_{Lim} separation distance and the closer limit, eq. 5.23, placed at 0.1 times this same distance which is a rough but valid enough approximation for the field sensed by a robot at the perimeter of a compact hexagonal cluster of less than 100 robots – a robot placed inside the cluster or the line will always sense a lower field module than those at the sides so it will consider itself inside the group–

$$\begin{aligned} Team_{Distant} &= Sg(x; x_{001}, x_{099}) \\ x &= \|\vec{\nabla}Team\| \\ x_{001} &= Team_{Far} \\ x_{099} &= Team_{Near} \end{aligned} \quad (5.22)$$

$$Team_{Near} = \cdot (0.1 \cdot n_{Sources}^2 \cdot r_{Lim}) \quad (5.23)$$

$$Team_{Far} = (n_{Sources}^2 \cdot r_{Lim}) \quad (5.24)$$

The descriptive element defined in eq. 5.22 can now be used to build the situations and their associated actions, that will guide the robot towards the group and the group towards the goal.

The first of these situations has already been introduced but now it is formalized. When the group is gathered so the robot is inside a the group, situation $S_{3.1}$ expressed in eq. 5.25, the group should to move towards the goal, so each robot in the group must move towards the goal. The action to accomplish each robot movement towards the goal is stated in table 5.7

$$S_{3.1} = \sqrt{1 - Team_{Distant}} \quad (5.25)$$

Table 5.7: Actions related with situation $S_{3.1}$ for groups

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	θ_{GOAL}	$\pi/2$	0	$S_{3.1}$

The complementary case, when the group is scattered or the sensing robot is out of the group, is split in several situations. The splitting is based on the relative positions of the group and the goal with the sensing robot so different actions can be taken for their different relative positions to be adapt to the manoeuvring limitations of the robots. There is no need for these situations to be perfectly delimited, the situations can overlap partially and the actions will be merged by the application of the proposed blending method.

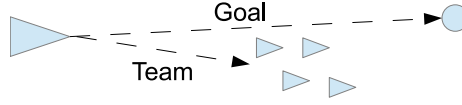


Figure 5.14: Situation $S_{3.21}$ depiction, when team and goal directions are aligned, the robot is at the back of the group

To describe the relative positions of the group and the goal with the sensing robot three more descriptive elements have been defined, $Team_{AlignedWithGoal}$, $Team_{OpposedToGoal}$ and $Team_{SideOfGoal}$ which help to build the three associated situations and their specific actions.

Taking the goal direction as the scenario global 'front', when the team gradient and the goal gradient were pointing to the same direction, the sensing robot is considered at the back of the formation –since the formation will be pointing towards the goal–. A descriptive element is then introduced to highlight when the goal and the team gradient were aligned, equation 5.26; this descriptive element follows the same idea used in the the case of the obstacle in the same direction of the goal on equation 5.7.

$$Team_{AlignedWithGoal} = \frac{1 + \cos(\theta_{Team} - \theta_{GOAL})}{2} \quad (5.26)$$

A situation can be defined now, which considers when the sensing robot is at the back of the group and the group is distant as depicted in figure 5.14. One more descriptive element is included, that there is no obstacle nearby. The inclusion of the obstacle consideration shows how a precedence or *subsumption* can be included within the situations prioritizing one over another, in this case the obstacle avoidance over the group gathering. The situation is fully expressed in equation 5.27 and the associated action, to move straight towards the group –and the goal– which is done through an Elect command centred in the team direction, can be found in table 5.8.

$$S_{3.21} = Team_{AlignedWithGoal} \cdot Team_{Distant} \cdot (1 - Obs_{Near}) \quad (5.27)$$

Table 5.8: Actions related with situation $S_{3.21}$ for groups

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	θ_{TEAM}	$\pi/2$	0	$S_{3.21}$

The opposite situation, when the sensing robot is between the group and the goal, as depicted in figure 5.15, and the group itself is far from the sensing robot, needs a different approach. Given that the considered robots only move forwards and turning around is costly, and since the group –their individuals– will move towards the goal due to $S_{3.1}$ and $S_{3.21}$, the easiest solution is to align the robot in the direction of the goal and to decrease the speed while the group is faraway, waiting for the rest of the team to get near. The descriptive element

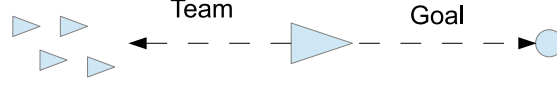


Figure 5.15: Situation $S_{3.22}$ depiction, when team and goal directions are opposed, the robot is ahead of the group

to establish when the robot is placed between the group and the goal is built in equation 5.28. The expression for the described situation is noted $S_{3.21}$ and can be expressed in equation 5.29. The expressions for the associated actions, point to the goal and decrease the speed while the team is faraway, can be found in table 5.9

$$Team_{OpposedToGoal} = \frac{1 - \cos(\theta_{TEAM} - \theta_{GOAL})}{2} \quad (5.28)$$

$$S_{3.22} = Team_{OpposedWithGoal} \cdot Team_{Distant} \cdot (1 - Obs_{Near}) \quad (5.29)$$

Table 5.9: Actions related with situation $S_{3.22}$ for groups

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	θ_{TEAM}	π	0	$S_{3.22}$
$B(\mu, \sigma, \delta, w)$	θ_{TEAM}	$\pi/2$	0	$S_{3.22} \cdot Team_{Distant}$

The last considered situation for the group gathering process is when the sensing robot observes the group at a side of the goal direction. This situation can be dealt with by moving the robot in a convergent direction with the group movement, a point which is in a direction somewhere between the group direction and the goal direction because the move will also move toward the goal during that time. This situation, $S_{3.23}$ is expressed in equation 5.31 through the use of the descriptive element for the relative positions expressed in equation 5.30. The robot movement is set for the middle direction between the goal direction and the group direction by an Elect action as expressed in 5.10

$$Team_{SideOfGoal} = \sqrt{|\sin(\theta_{TEAM} - \theta_{GOAL})|} \quad (5.30)$$

$$S_{3.23} = Team_{SideOfGoal} \cdot Team_{Distant} \cdot (1 - Obs_{Near}) \quad (5.31)$$

Table 5.10: Actions related with situation $S_{3.23}$ for groups

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	$\frac{\theta_{GOAL} + \theta_{TEAM}}{2}$	π	0	$S_{3.23}$

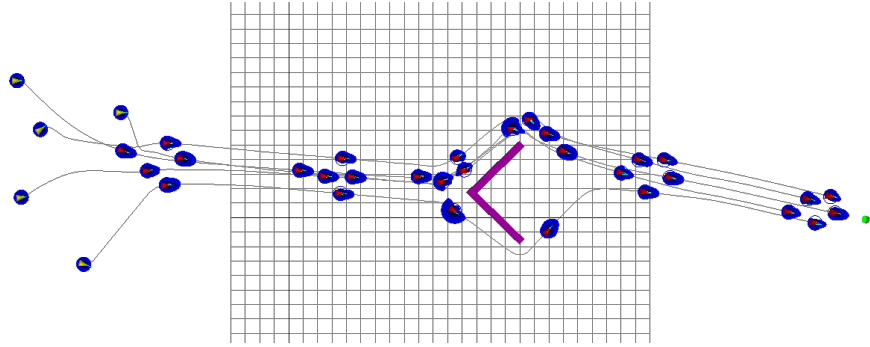


Figure 5.16: Simulation captures along the path for an unstructured robot group

5.3.2 Simulation

The simulation testing of the described system is now shown, where a group of five robots, initially dispersed with random orientations, aim for a single goal. The presence of an obstacle in the group path is intended to disrupt the group to show the robots' behaviour in a group disruption. Figure 5.16 shows the individual robots' path along the test with captures of the whole set of robots at several moments of the simulation. Initially it can be seen how all the robots are far apart from each other and not facing the goal. As the simulation begins a cluster of robots is quickly formed which travels together towards the goal. When the obstacle is reached, its geometry forces one robot to move apart from the main group while the group continues together along the obstacle. Once the obstacle is surpassed the robot cluster is built again, it can be seen in the figure how the detached robot's direction is that between the goal direction and the group direction angle when moving towards the main cluster of the group. But also it can be observed how the main cluster of the group moves towards the detached robot and not directly towards the goal, however the group's movement towards the detached robot has less emphasis than the opposite, but the group's movement does not leave back the detached robot. With the group complete again, the movement is straight to the goal. All the scenario runs without collisions even in the proximity of the obstacle, where the robots should take care of both the obstacle and the other robots and near the goal, where the goal direction for each robot can no longer be considered parallel.

Studying the situation indicators for all the robots present in the test will be too extensive, but it is interesting to observe in detail at least the situations related with the team field in one of the robots. The richer situations can be found in the robot detached by the obstacle so in figure 5.18 the evaluation along the simulation of the four situations introduced for handling the group behaviour are shown. Also the value of the $Team_{Distant}$ descriptive element, which is the core of the group situations, can be seen in figure 5.17.

The detail on when the robot belongs and belongs not to the group along the path in figure 5.16 can be easily followed along the representation of the $Team_{Distant}$ value shown in figure 5.17. Initially the robot is not among the group, until $t \approx 20s$ when the $Team_{Distant}$ indicator decreases, meaning the

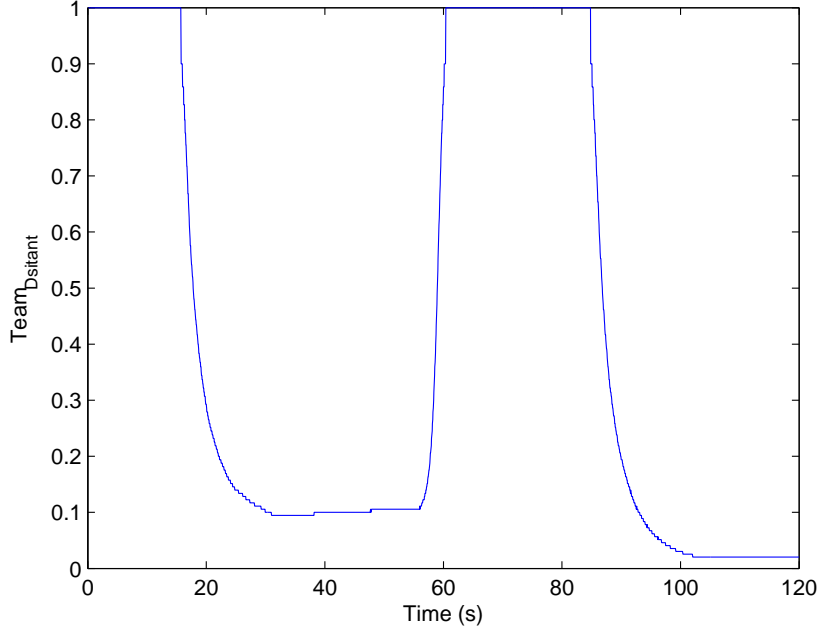


Figure 5.17: $Team_{Distant}$ descriptive element for the detached robot along the group simulation

robot is closer enough to the main group –when the robot are packet together, the equivalent team field source is placed among them–. The robot continues attached to the group until $t \approx 60s$ when the presence of the obstacle forces the robot to break out of the group. The robot remains detached from the group until $t \approx 90s$ when the obstacle is surpassed and it can gather again with the rest of the robots until the goal is reached. The low but higher than zero values of $Team_{Distant}$, even when the robot is among its peers in the cluster, at the first gathering is due to the group not being densely packed. In the second gathering at the end of the simulation time, the group is more densely packed so the $Team_{Distant}$ value is lower. However both values are low enough to render the gathering actions, $S_{3.2x}$, irrelevant while making dominant the goal seeking action associated to $S_{3.1}$. This is clearly shown at the situations figure, 5.17.

Having a look at the evolution of the situation estimators we can study the scenario evolution with a deeper understanding on the behaviours that guide the robots. The evolution of the robot detached from the group by the obstacle is which shows more events, so this robot will be used to illustrate the situation analysis. The situation estimators of the robot detached for the group by the obstacle are shown at figure 5.18. This robot is also the one initially located at the lower side of the image.

At the beginning of the simulation, the group center is at a side of the robot, and in figure 5.16 it can be seen how the robot moves diagonally to meet the group and place itself at the side and slightly at the front of the group. This side-front position can be appreciated through the situations because $S_{3.22}$ –being ahead of the group– and $S_{3.23}$ –being at the side of the group– values get closer around $t \approx 15s$ while $S_{3.21}$ –being at the back of the group– falls.

Even along $t \approx 20s$ to $t \approx 60s$, when the group is gathered and moving towards the goal because $S_{3.1}$ –the group is gathered– is the dominant situation, the values of $S_{3.22}$ and $S_{3.23}$ are quite similar and above $S_{3.21}$.

The appearance of the obstacle near $t \approx 60s$ drops all the situational values related with the group low because of the precedence of the obstacle avoidance behaviour term, $(1 - Obs_{near})$ introduced in the situation descriptors $S_{3.x}$.

Once the obstacle is clear, $t \approx 75$, the descriptors for distant group $S_{3.2x}$ jump up, and now they show the group completely at a side of the robot – $S_{3.21}$ and $S_{3.22}$ are both similar and much lower than $S_{3.23}$ –.

After a while, around $t \approx 100$, the group is gathered again, the distant group descriptors falls and $S_{3.1}$ becomes the dominant situation driving the robot towards the goal.

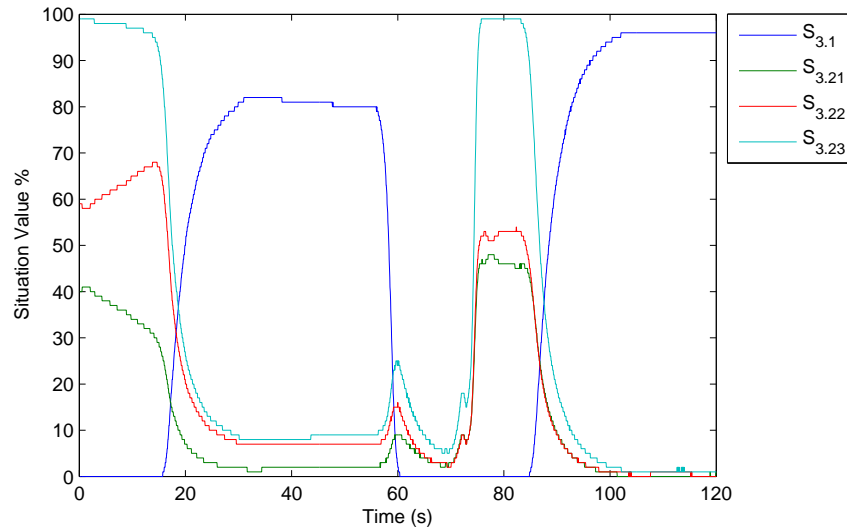


Figure 5.18: Group related situation indicators for the detached robot along the group simulation

5.4 Robot formations

The next step on the possible schemes that can be applied to the navigation of a group of robots is the formation navigation. The main difference between formations and groups is the level of structure found in both, while the group do not need any specific positioning of the robots, a formation requires at least some relative positioning among the formation individuals. Many kind of formations and formations structures can be devised but some elements are common for all the formations structures.

The local reactive nature of the proposed method and its application for local navigation impels the studied formation scheme to be decentralized; in the proposed scenario there is no coordinator nor any explicit coordination among the robots in the formation. The formation scheme is built up, as in the previous cases, with anonymous robots. There are not designed roles in the formation structure, the robots are completely interchangeable and no especial information is delivered to or used by specific robots. This characteristic is highly related with the use of virtual fields, where all the sources of a given field kind are considered equal and the information, once integrated inside the field, cannot be traced back to their specific sources. However, a formation where specific robots are attached to specific sections of the formation can also be achieved through the use of different field kinds, like with the two groups in the multiple independent group case seeking two different goals.

Finally, the formation shape can be defined in two ways as is shown in figure 5.19: through general traces or through dots, specific spots where the robots must place themselves. The use of general traces allows to more flexible formations because it does not limit the number of robots in the formation and allows the robots to better adapt to obstacles or any other incidence. However the placement of the robots is less accurate or symmetrical using general traces instead of dot structures. On the other hand, dot structures allow more detailed formation structures of any shape and with any pattern which cannot be achieved using single traces, but dot structures need to be modified to accept new individuals. When using non-holonomic robots, trace described formations can result in formations easier to achieve due to the higher flexibility that they offer. With the set of behaviours and actions described along this case both trace described and dot described formations can be achieved –actually also mixed structures can be built–. The greater emphasis is placed in the trace described formations, while the dot described formation are shown in less detail.

In the presented navigation scheme, each individual robot seeks its own way to build the formation structure. Along the process there is no communication to coordinate the robots other than the periodical broadcast of each robot telling its position. Only this position information is used to build the virtual fields associated to the robots and is the same already used along the previous cases.

5.4.1 System Design

Following the previous ideas to hold system information within virtual fields, the formation shape is built using a virtual field. This formation fields are built in a way that the minimum of the field follows the formation trace, as it is illustrated in figure 5.20. The formation field itself can be build in many ways, for simplicity mathematical expressions are used for the current study, but also

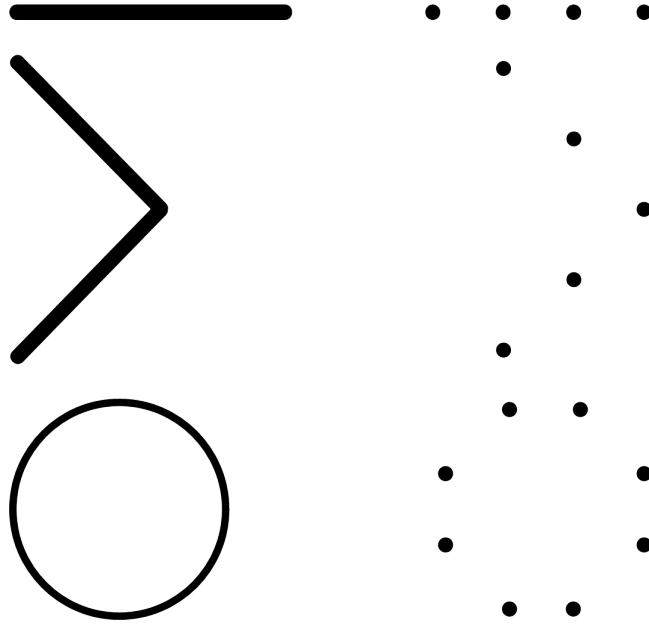


Figure 5.19: Formations defined through traces vs. formations defined by dots

height maps or other means can be used to build the formation field as long as it can provide values for the potential and field gradient at any requested point. There is one limitation to the formation shapes defined only through a trace as they are used here: the formation field must come from a single continuous trace, meaning that a formation of two parallel lines will not be valid. Aside from this, any single continuous trace, closed or not, with vertex, undulated, etc can be used as formation shape with the the proposed set of situations and actions.

Formation field The formation field is the entity that holds the information about the desired shape to be acquired by the set of robots. In a very explicit way the formation field will adopt the shape of the desired formation and the robots will move towards the region of minimum potential of the formation virtual field.

The situations and actions described along this section are intended to be general enough to work with most field shapes. Actually the situations do not relay on specific knowledge of the formation shape, just on the formation potential value and its gradient at the current robot location, in the same way used on all the previously presented cases.

The general idea behind the proposed set of situations is to guide the robots towards the formation potential minima region and once there move along the formation movement direction without leaving the minima region; using this approach there is no need to know the formation shape, just the direction of the minima region, only the gradient direction.

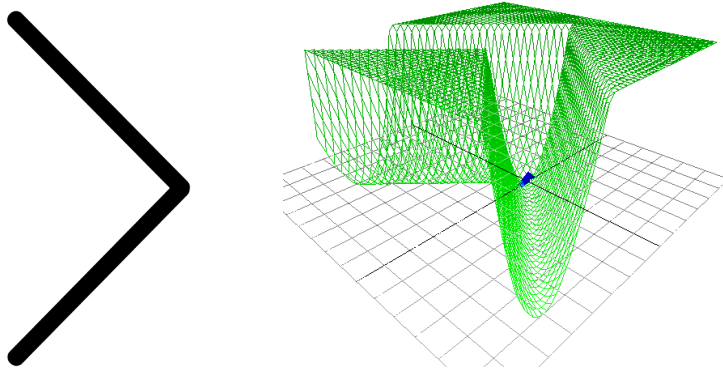


Figure 5.20: Formation field built from a formation trace

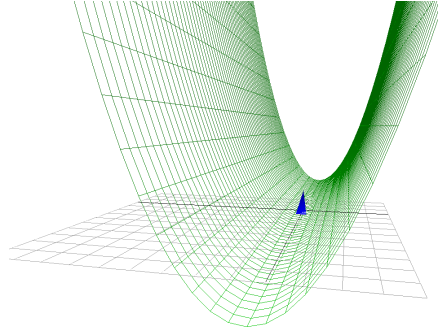


Figure 5.21: Line formation field

In order to test this generalist approach several field shapes were employed along the test, while the number of possible formation shapes is countless, a set of common representative formation shapes were chosen. The formation field expressions are built on its own reference system where x represents the formation movement direction and y its perpendicular. Some other formation fields, mainly variations of the previous ones, have been studied also while fitting the method and situations, however the ones here shown are the most representative ones.

- Line formation, where each robot is aligned with its mates in a single line in the formation movement direction, built through the expression in equation 5.32 and represented in figure 5.21

$$\phi_{FRM_{Line}} = y^2 \quad (5.32)$$

- Column formation, where each robot is aligned with its mates in a single line perpendicular to the formation movement direction, it is built using equation 5.33 and shown in figure 5.22

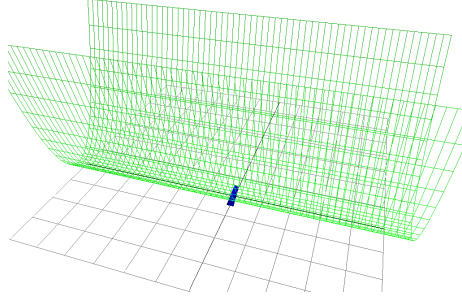


Figure 5.22: Column formation field

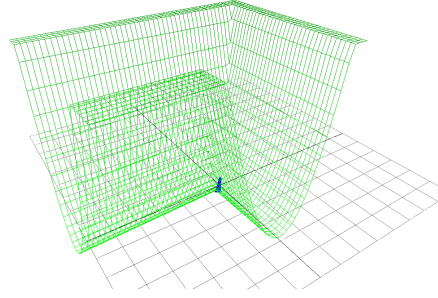


Figure 5.23: Wedge formation field

$$\phi_{FRM_{Column}} = x^2 \quad (5.33)$$

- Wedge formation, where the robots are placed along two lines which are not aligned, nor perpendicular, to the formation movement direction and join in a vertex at the front of the formation, equation 5.34 and figure 5.23

$$\phi_{FRM_{Wedge}} = (x + |y|)^2 \quad (5.34)$$

- Circle formation, where the robots are distributed along a circle and each neighbour has a different relation between the formation movement direction and the formation trace at its location, expressen in equation 5.35 and depicted in 5.24

$$\phi_{FRM_{Circle}} = (r - r_0)^2 \quad (5.35)$$

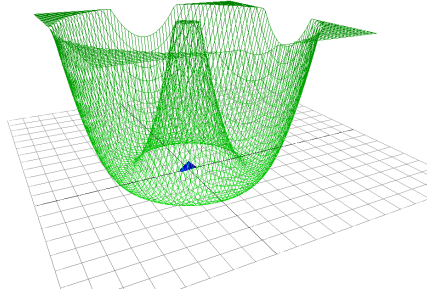


Figure 5.24: Circle formation field

Formation field positioning In the previous studied cases, the level of structure was low enough so no common knowledge has to be shared among the robots. In the case of the unstructured robot group the 'group' entity was not defined as a single element but built out from the addition of the group field being sourced by the individual robot locations.

In the case of a formation field, the field itself cannot be built out of the addition of several sources since it should present a specific shape that is easier to describe using a single field source, therefore an independent formation field is used, and its source is not attached to any specific robot.

To place this formation field in the scenario, a specific mechanism have be designed to establish where the formation field must be positioned among the robots. Using this mechanism, each robot places its own virtual formation field according to its own data about the position of the other robots belonging to the formation. The small discrepancies between the real robot positions and the positions used by each robot on the placing of the formation field are filtered by the iterations on the positioning algorithm.

To ease its description, the formation field is described in its own coordinate system, so the coordinate origin is chosen only by convenience on the trace expression. In a wedge formation field the more natural position for the coordinate origin is the vertex of the wedge, while in the circle formation field the natural place to set the origin in the the center of the circle. The coordinate origin for the virtual field is therefore not related with any characteristic, like the center of mass, mode or median position, of the collection of robots.

In order to keep the robots moving in the desired shape, the formation field needs to be moved along with the robots. A variation of a gradient descent method had been employed for this.

Initially, when the formation field is first applied, the coordinate origin of the formation field is placed at the centroid of the robots, and the formation field x axis aligned with the desired movement direction, usually set in the goal direction.

After the first positioning the formation field origin is moved –translation only, not rotation– in a direction that reduces the the mean of the formation

gradient module of all the robots.

This new relative position of the virtual formation field origin with the robots centroid is then remembered to be used as initial position for the next correction step.

As the robots move, the position of the virtual formation field coordinate origin will be the robots centroid position plus the previously reached relative position, at each control iteration new corrections are made in the same way to reduce the mean formation gradient module.

These iterative corrections result in that the formation field is kept in a position relative to the centroid of the robot positions that minimizes the mean of the formation gradient of the robots.

The specific expression used in the algorithm implementation can be found in equation 5.36 where $F\vec{pos}(n)$ is the position of the formation coordinate origin in the world coordinates, \vec{C} is the centroid of the robots, \vec{E} is the mean of the formation gradient at the robot positions and \vec{E}_j are previous values of \vec{E} .

The values of K_p and K_i are related with the dynamics of the robots moving as a formation and represents the 'inertia' or 'attenuation' of the formation response to robot group disorders. For the employed robots, values of $K_p = 0.05$ and $K_i = 0.01$ have been used, however those values are not critical and have not been optimized.

$$F\vec{pos}(n) = \vec{C} + K_p \vec{E} + K_i \sum_{j=0}^n \vec{E}_j \quad (5.36)$$

This positioning scheme allows the use of formation shapes where the robots centroid, when a stable formation is reached, depends on the number of robots involved. This allows to place the formation field coordinated origin at any convenient location.

In some circumstances, when sideways motion of the formation is not desired, the same formation placement scheme, described above, can be used with the variation of projecting \vec{E} only on the movement direction, θ_{LEAD} . By doing this, the formation center moves only along the movement direction; this is a simple way to keep formation structure along a specific path even when the robots do need to move to a side due to the presence of some obstacle.

While this basic scheme works good enough most of the time, using the mean of the formation gradient at the robot positions, sometimes it results in a formation that takes a long time to stabilize or even it can't reach stability. This happens when one robot, or a small portion of the robots, is blocked far from the rest of the group. The presence of a far robot, specially if it is at the back blocked by some obstacle, results in a constant drag of the formation structure so it cannot fit with the remaining robots.

To avoid this, the contribution to \vec{E} of each robot is weighted comparing the individual field module measured at the robot location, e_i , against the mean module, $\|\vec{e}\|$, and standard deviation, $\sigma \|\vec{e}\|$, of the complete collection of local measurements. The weight function takes only full weight (0.99) for those robots gradient values under $\|\vec{e}\| \pm \sigma \|\vec{e}\|$ or under $\|\vec{e}\| \pm r_{Lim}$ if the value of $\sigma \|\vec{e}\|$ is smaller than r_{Lim} . On the other side, there is almost no contribution (0.01) of

those robots with gradient values greater than $\|\vec{e}\| \pm 3 \cdot \sigma \|\vec{e}\|$ or greater than $\|\vec{e}\| \pm 3 \cdot r_{Lim}$. The specific weight value is obtained through the sigmoid functions stated in eq. 5.37

$$w_i = \begin{cases} Sigmoid(\|\vec{e}_i\|, \sigma \|\vec{e}\|, 3 \cdot \sigma \|\vec{e}\|) & \sigma \|\vec{e}\| > r_{Lim} \\ Sigmoid(\|\vec{e}_i\|, r_{Lim}, 3 \cdot r_{Lim}) & \sigma \|\vec{e}\| < r_{Lim} \end{cases} \quad (5.37)$$

The value of \vec{E} actually used in 5.36 is built out of these weighted contributions, as described in 5.38

$$\vec{E} = \frac{\sum w_i \cdot \vec{e}_i}{\sum w_i} \quad (5.38)$$

Goal and Leading direction In the previous cases, robots aim always towards a goal and they use the goal field as a movement direction reference. In the present case the robots themselves do not aim towards the goal, but it is the formation structure the one that aims toward the goal. The goal field is also present along the current case but it is now used to point the formation in the goal direction; the robots never use it. In the formation field positioning process, the formation field coordinate system is aligned with the goal direction using the goal field at the position of the formation field coordinate origin. Since usually the robot is not placed in the formation field coordinate origin, the goal direction measured there and that measured at the robot position will not necessarily be the same.

Using the goal direction at the formation field coordinate origin as movement direction reference, all the robots shares the same aiming direction.

To avoid confusions and ease the readability, the goal direction at the formation field coordinate origin will be referred as the *Leading direction* and this term, somehow expressed in equation 5.39, is the one that will be used along the situations and actions definition.

$$\theta_{Lead} = \theta_{Goal} \left(F \vec{Pos} \right) \quad (5.39)$$

No explicit situation or action is set to guide the robot towards the goal or the *Leading direction*, the movement of the robot in the *Leading direction* will be set only in combination with the formation field. Since the robots do not aim directly towards the goal, the situation $S_{1,0}$ described in eq. 5.8 and their associated actions found in table 5.1 are not considered for this case.

Obstacles Being not the goal direction the robot target direction, the obstacles are considered following the descriptive elements Obs_{Near} and Obs_{InPath} that can be found in equations 5.3 and 5.18 on the unstructured robot group case. The obstacle warning situation, $S_{1,1}$, is also the same used in the unstructured robot group case and expressed in equation 5.19 with their associated action found in table 5.2.

Other robots In this case all the robots present on the scenario are assumed to belong to the formation, other robots can be present but for coexistence they should use a different set of fields to avoid conflicts. For those robots on the same formation, to avoid robot-robot collisions, the set of situations and actions introduced in the unorganized group case are taken. The robot

proximity warning is handled by situation $S_{2.1}$ as defined in equation 5.14 and their associated set of actions to avoid the collision defined in table 5.6. Again, as in the unorganized groups, there is no need to actively skirt around near robots since they belong to the same group, so situation $S_{2.2}$ defined in equation 5.15 and their associated actions are not being used.

However, to ease the manoeuvres of the robots into the formation, one more descriptive element is employed along this case. To keep the robots far enough from each other and provide them with manoeuvring space, the descriptor D_{RM} is introduced at equation 5.40. Its expression is analogous to Rob_{Near} , in equation 5.11, but it expands the collision avoiding space farther, to allow close manoeuvring without entering collision space.

$$\begin{aligned} D_{RM} &= Sg(x; x_{001}, x_{099}) \\ x &= 1/\sqrt{\|\Phi_{Rob}\|} \\ x_{001} &= 3 \cdot r_{Lim} \\ x_{099} &= 5 \cdot r_{Lim} \end{aligned} \tag{5.40}$$

Formation In the study of previous works related with formations several steps can be identified on the formation navigation. Some works like care mainly to the navigation process once the formation is already accomplished, testing the manoeuvrability and stability of the formation, while other works are mainly centred on the accomplishment of the formation from random positioned robots.

To easy the handling of the situations related with the formation navigation, the situations have been grouped along three different stages of the formation process:

- The robot is far from the formation and it needs to get near.
- The robot is near the formation but it needs to manoeuvre around it in order to find a place inside the formation.
- The robot is inside the formation structure and it must move coherently with its formation partners in order to reach the goal.

Certain number of descriptive elements are so introduced to handle formation stages and the relative positions and orientations with other elements of the scenario which are needed to define the situations.

Formation descriptive elements To establish the three stages of the formation gathering process a descriptive element is introduced to keep track of the distance between the robot and the formation based on the measurement of the formation field. Two inflexion points are defined for this descriptive element: d_{FI} sets the distance where the robot is considered near enough to get inside the formation and d_{F0} establishes where the robot can consider itself inside the formation. In the space between d_{FI} and d_{F0} will take place the insertion manoeuvres where the robots need to move around the formation in search for an empty place while avoiding collisions with other robots and trying not to disturb too much the existing formation. Those limit points, built in regard of the manoeuvring capabilities represented by r_{Lim} , are stated in equations 5.41 and 5.42. The Descriptive element, D_{FI} , built from this two saturation points is defined in eq. 5.43. At distances greater than d_{FI} the robot can be considered

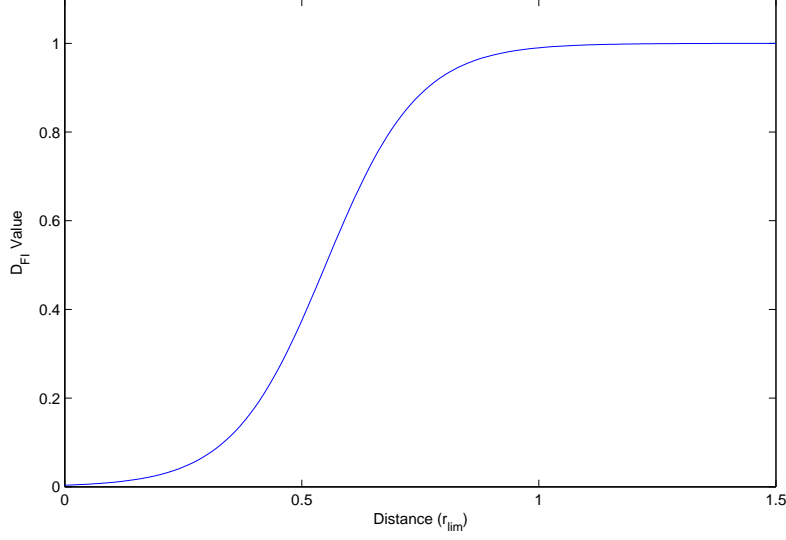


Figure 5.25: Normalized descriptor for formation distances

far from the formation and the value of D_{FI} will be high while when the robot is inside the formation D_{FI} will have a near zero value as is represented in Figure 5.25.

$$d_{FI} = r_{Lim} \quad (5.41)$$

$$d_{F0} = \frac{r_{Lim}}{10} \quad (5.42)$$

$$\begin{aligned} D_{FI} &= Sg(x, x_{001}, x_{099}) \\ x &= \|\nabla \phi_{FORM}\| / 2 \\ x_{001} &= d_{F0} \\ x_{099} &= d_{FI} \end{aligned} \quad (5.43)$$

The positioning of the elements around the sensing robot, and its own location relative to the formation/group/etc, have been built around the leading direction. As in the case of the unorganized robots, the actions needed for a robot to approach and get inside a formation can be better tuned by considering the position of the sensing robot with regard to the formation and the leading direction.

The manoeuvres depend more on the sensed formation direction relative to the known leading direction rather than on the formation direction alone. To handle this, the building of the Descriptive Elements related with the position of the formation and other robots uses the leading direction as a reference. In this way, the the leading direction defines the 'front' side of the scenario.

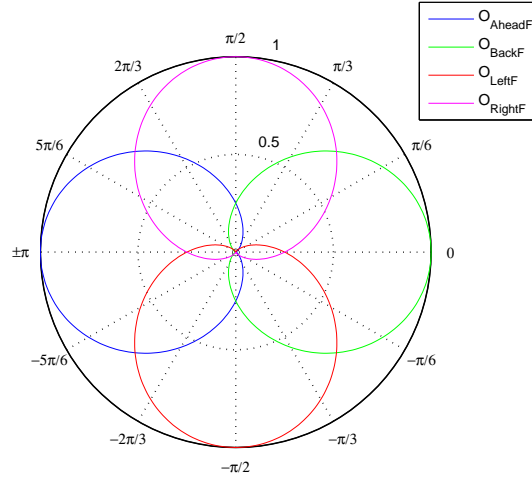


Figure 5.26: Relative position of the robot within the formation

The position of the sensing robot relative to the formation is established through the descriptive elements defined in equations 5.44 to 5.48, where the front side of the formation is the one in the leading direction, in a similar way that the team relative positions were defined in the case of the Unorganized Robots Group in eqs. 5.26, 5.28 and 5.30.

When θ_{FORM} and θ_{LEAD} are aligned then both the formation and the goal are in the same direction for the sensing robot, so the robot itself is placed at the back of the formation. When θ_{FORM} and θ_{LEAD} are opposed, the goal is at one side of the robot and the formation at the other one, so the robot must be ahead of the formation. The coverage of those descriptive elements is shown in figure 5.26

The last element, introduced in eq. 5.48, O_{FSide} normalizes the side of the formation relative to the leading direction between ± 1 .

$$O_{AheadF} = \left(\frac{1 - \cos(\theta_{FORM} - \theta_{LEAD})}{2} \right)^2 \quad (5.44)$$

$$O_{BackF} = \left(\frac{1 + \cos(\theta_{FORM} - \theta_{LEAD})}{2} \right)^2 \quad (5.45)$$

$$O_{RightF} = \left(\frac{1 + \sin(\theta_{FORM} - \theta_{LEAD})}{2} \right)^2 \quad (5.46)$$

$$O_{LeftF} = \left(\frac{1 - \sin(\theta_{FORM} - \theta_{LEAD})}{2} \right)^2 \quad (5.47)$$

$$O_{FSide} = \sin(\theta_{FORM} - \theta_{LEAD}) \quad (5.48)$$

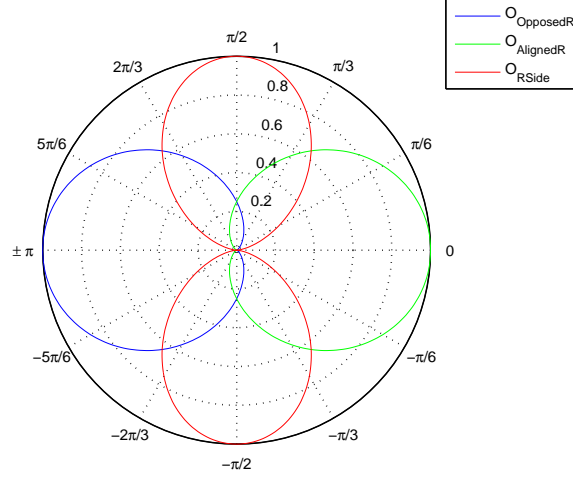


Figure 5.27: Descriptive elements for sensed robot position relative to leading direction in polar representation

In a similar way, the position of the sensing robot regarding the other sensed robots related to the leading direction is handled by the descriptive elements found in equations 5.49 to 5.51, this can be seen at figure 5.27

$$O_{AheadR} = \left(\frac{1 + \cos(\theta_{ROB} - \theta_{LEAD})}{2} \right)^2 \quad (5.49)$$

$$O_{BackR} = \left(\frac{1 - \cos(\theta_{ROB} - \theta_{LEAD})}{2} \right)^2 \quad (5.50)$$

$$O_{RSide} = \sin(\theta_{ROB} - \theta_{LEAD}) \quad (5.51)$$

One more descriptive element needs to be introduced, $O_{AlignedRF}$ in equation 5.52, which establishes when the *equivalent robot* is in the direction of the formation, so there is one or more robots blocking the straight path towards the formation.

$$O_{AlignedRF} = \left(\frac{1 - \cos(\theta_{FORM} - \theta_{ROB})}{2} \right)^2 \quad (5.52)$$

Situations As the complexity of the scenario grows, some general ideas about the whole movement of both, robots and formation, need to be established before formalizing the situations to ensure the coherence of the final system.

- When the robot is far from the formation, the main objective of the robot is to get itself near the formation and then inside it.
- When a robot is inside the formation, its main task is to keep the formation and to move it in the leading direction.

- When a robot is not inside the formation and moves to get inside, it must assume that the formation itself is also moving.
- The robots inside the formation needs to consider that the robots outside the formation have to be able to move inside the formation.
- An arrangement needs to be achieved between the speed of the formation and the speed of the individual robots. A limit on the formation movement speed should be established to allow the individual robots to reach the formation in a reasonable time.

The previous considerations establish also the existence of two differentiated behaviours of the robots: one when the robots are far for the formation and the other when the robots are inside the formation. Two subsets of situations are therefore considered for the building of the situations and actions.

The first subset of situations-actions handles the behaviour of the robot when it is far from the formation. This situation is defined simply through the use of D_{FI} as shown in equation 5.53.

$$S_{3.1} = D_{FI}; \quad (5.53)$$

The main objective of the robot when it is far from the formation is to move towards the formation considering that the formation is also moving in the leading direction. To handle this, two actions are defined: to elect the leading direction with a partial deviation towards the formation and to brake in the leading direction when the robot is ahead of the formation. Those two actions are conditioned by the path towards the formation being free of other robots, if other robots are present blocking the path, the actions will be modified accordingly. The Elect direction will be just the leading direction and the Brake intensity will decrease.

Table 5.11: Approaching Actions $S_{3.1}$ for formations

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	μ_E	$\pi/2$	0	$S_{3.1}$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3.1} \cdot O_{AheadF} \cdot freePath$

Where:

$$\begin{aligned} \mu_E &= \theta_{Lead} + \pi/5 \cdot O_{FSide} \cdot freePath \\ freePath &= (1 - O_{AlignedRF} \cdot (1 - D_{RM})) \end{aligned}$$

The set of actions defining this behaviour can be found in Table 5.11. In 5.11, the central elected direction, μ_E is built so it aims first to the leading direction, the second additive term introduces a deviation towards the formation through the use of O_{FSide} , which has a value between +1 and -1 depending on the side of the formation in which the robot sensing is located. This deviation from the leading direction is limited to $\pm\pi/5$, this limit is reached when the formation direction is perpendicular to the leading direction. Through this limitation the robot will always move at least $\cos(\pi/5) \cdot r_{vel} \approx 0.8 \cdot r_{vel}$ in the leading direction, where r_{vel} will be the instant speed of the robot at that moment.

Using this we can establish the the maximum speed allowed to the formation to $0.8 \cdot v_{Max}$ which will allow a robot situated at a side of the formation to reach the formation in any case. Finally, the *freePath* term just expresses if there is a robot in the path towards the formation, and can be read as : *Not(Robot_and_Formation_Aligned And Closer_than_Robot_Manouvering_Distance)* . This term is introduced to avoid a potentially jamming where a robot tries to move towards the formation but another robot is already there, preventing the first to move closer to the formation. In the case where there is another robot in the direction of the formation, the first robot will move along the leading direction passing out the second robot and when the path to the formation is clear enough turn again towards the formation. The weight term in the Brake action just states that the robot must brake when it is ahead the formation and there is a free path towards the formation –there is not another robot close to its back–.

To avoid an special symmetry case, another action is also considered when the robot is far from the formation.

This action is specially crafted to cover when two robots are approaching a line formation from different sides of the formation trace and aiming for the same spot on the formation. In that situation the two robots will move towards the leading direction with $\pm\pi/5$, as ruled by the actions in Table 5.11.

When the two robots are out of the formation and at equal distances of the formation trace, each at different side of it, their approaching angle and speed will be the same. Following the current approach actions, both robots will reach a mutual distance where the collision avoiding actions will prevent them to move closer to the formation, so they will evolve to move in parallel to the formation and to each other without being able to move closer to the formation.

To avoid this, a brake action expressed in Table 5.12 is added to force one of them to brake while the other not. This inequality is enough to avoid the jam and does not affect other situations. The expression in 5.12 will apply the Brake action in the leading direction when the robot is at the right side of the formation and there is another robot in the formation direction. In this way the robots in the left will have preference over the robots at the right, which will brake, an let the formation move ahead in search of a free spot into it.

Table 5.12: Special Symmetry avoidance in $S_{3,1}$ for formations

Distribution	μ	σ	δ	w
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/4$	0	$S_{3,1} \cdot O_{RightF} \cdot \sqrt{1 - freePath}$

When the robot is getting inside, or is already inside the formation, the movement objective changes, the formation is already reached, and now the formation needs to be kept in shape and moving in the leading direction. This situation, being in the formation, is expressed in eq. 5.54 and a collection of actions are taken to maintain the formation in shape, accept more robots inside it and keep it moving in the leading direction.

$$S_{3,2} = (1 - D_{FI}); \quad (5.54)$$

To keep the formation in shape and the robots inside it, three actions found

in table 5.13 are taken,

- To favour the movement direction that reduces the formation gradient, but keeping the leading direction as the main movement direction.
- To brake in the leading direction, proportionally to the gradient module, when the robot is ahead the formation.
- To set a constant slight brake action –unless there is another robot too close at the back– to prevent the robots inside the formation from going too fast, to allow outside robots to reach the formation .

Applying these actions always, prevents the robot from driving out of the formation.

Table 5.13: Formation keeping actions related with situation $S_{3.2}$ for formations

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	μ_E	$\pi/2$	0	$S_{3.2} \cdot O_{AlignedRF} \cdot (\frac{1}{2} + \frac{1}{2}D_{RM})$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3.2} \cdot O_{BackF} \cdot D_{RM} \cdot \ \vec{\nabla}\phi_{Form}\ \cdot 0.1$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3.2} \cdot (1 - O_{BackR} \cdot (1 - D_{RM})^2) \cdot 0.2$

Where:

$$\mu_E = \theta_{Lead} + \pi/5 \cdot O_{FSide} \cdot O_{AlignedRF} \cdot D_{FI}$$

A second set of actions, stated in Table 5.14, is considered to ease the insertion of robots from outside the formation by those robots already inside it. The robots already inside the formation are set to move slightly to the side opposite to the outside robot direction or to brake a little when the outside robot is ahead, always applying this movements without getting out of the formation. These actions are enough to provide a void place in the formation for the incoming robot to get inside it.

Table 5.14: Formation insertion actions related with situation $S_{3.2}$ for formations

Distribution	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	μ_E	$\pi/2$	0	$S_{3.2} \cdot (1 - D_{RM}) \cdot 0.5$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3.2} \cdot (1 - D_{RM}) \cdot O_{AheadR} \cdot 0.6$

Where:

$$\mu_E = \theta_{Lead} + \pi/5 \cdot O_{RSide} \cdot D_{FI}$$

5.4.2 Simulation

The previous situation-actions scheme is here tested through simulation. In the previous cases, the simulation was used mainly to illustrate the evolution and results of the applied method and how the proposed situations affect the results.

Being the formations among the most complex scenarios, the formations scenarios are used for the validation of the proposed method. A more extensive

study is done in simulation, here, and with real robots, in the next chapter, where both results are compared.

The next simulations will study the proposed situation-action scheme for different formation shapes and different number of robots. Some parameters, related with the evolution of any formation structure and not with the method, are studied for each combination of formation shapes and robots in them. By the obtention of coherent results along tests with different formation shapes and number of robots, we expect to show the viability of the presented method to handle complex scenarios through situation-actions schemes beyond those directly shown here.

Formation Initialization

Formation initialization is probably the most crucial stage in the achieving of a stable robot configuration with a given formation shape. The reason is that any time at which the robots are not within a stable formation can be considered as an initial position. For a method of reactive nature, as the proposed one, an extensive study of the initial cases will cover most of the casuistic that could appear on any other situation.

Robot Movement Examples Several combinations of formation shapes, number of robots and initial positions of the robots are going to be studied along this section and several of them will be accompanied by snapshots of the simulation evolution to illustrate how the robots behave. To homogenize the view of this snapshots, they have been always arranged in the same way.

The first frame, top left, shows the initial position of the robots, just before the simulation begins to run.

The second frame, top right, shows the initial movement of the robots towards the formation. In that second frame the robots barely move closer to the formation than they were in the first frame, however they oriented themselves towards the formation, which is the first needed step being non-holonomous robots.

In the third and fourth frames, middle left and middle right, it is shown the travel done by the robots towards the formation. There, the formation shape begins to be built by some robots, which were the closer ones to the formation at their initial positions. While the others have to travel from their initial positions to a place within the formation.

Of the two final frames, the fifth frame, bottom left, shows the formation almost completed, most of the robots are already inside the formation, the formation shape is clearly recognisable and only adjustment manoeuvres of the last robots are needed to achieve the final stable formation. Finally, the bottom right frame shows the stable configuration of the robots within the given formation shape.

On all simulations shown, the leading direction is in the X direction, the right side of the frames.

Line formation Along figures 5.28 to 5.31 a sample of the achieving of a line formation using 5, 7, 11 and 20 robots is shown.

The first frame of figure 5.28 shows the initial position of the five robots taking part in this simulation, initial positions and orientations of the robots are randomly generated.

In the second frame of Figure 5.28 it can be seen the initial turn direction taken by the robots when the formation building process begins. Along this first frame most of the robots are not near enough to other robots so the only influence is the formation field, therefore the robots turn to reach an approaching direction.

In the third frame is quite visible the non-holonomic nature of the robots as they all turn towards the formation with a non zero radius arc. Also at this frame the two robot nearest to the formation reach it.

On the fourth frame, three robots have reached the formation and the formation shape is getting clear. At this fourth frame the approximation angle defined at the *Elect* action in $S_{3,1}$ can be appreciated. The robots do not move straight towards the formation but with a given angle between the leading direction and the formation, equal for all the robots approaching to the formation since all of them sense the same formation field and leading direction. This angle gets softer and aligns with the leading direction as the robot moves closer to the formation and becomes under the influence of $S_{3,2}$ actions.

In the fifth frame of figure 5.28 nearly all the robots are inside the formation shape but the formation is not completely done yet, since the distance between robots is not enough and the final alignments with the leading direction needs to be fit. Finally the sixth frame shows the final stable formation: all the robots within the formation shape, far enough one from another and all travelling in the leading direction.

The process taking place in figure 5.29, 7 robots in a line formation, is mainly the same as in figure 5.28, however since there are more robots involved, the robot-robot interferences grow up.

In the fourth frame it can be seen how the two robots at the end of the formation are aiming to the same place inside the formation. That illustrates the situation covered by the action shown in table 5.12; to avoid a jamming in the formation building process one of the robots needs to have precedence over the other one.

It can be seen in the fifth frame how the rightmost robot at the end of the formation in the fourth frame, even being the first to reach the formation inner space, provides space to the other robot by reducing its speed, and the left most robot at the end of the formation in the fourth frame is keeping its speed and moves skirting the other robot, ending by placing itself ahead of it.

In the last frame of figure 5.29 the situation is solved and the formation have reached the stable configuration.

In figure 5.30, 11 robots in line formation, the robot-robot interference grows as the number of robots is increased again.

The first three frames evolve mostly equal than in 5.28 and 5.29, the robots move toward the formation line with the limit approaching angle established in the actions for 3.1, while solving easily the collisions just by moving to a side.

It is in the fourth frame were several cases of robots trying to reach the same positions or positions already occupied within the formation can be observed.

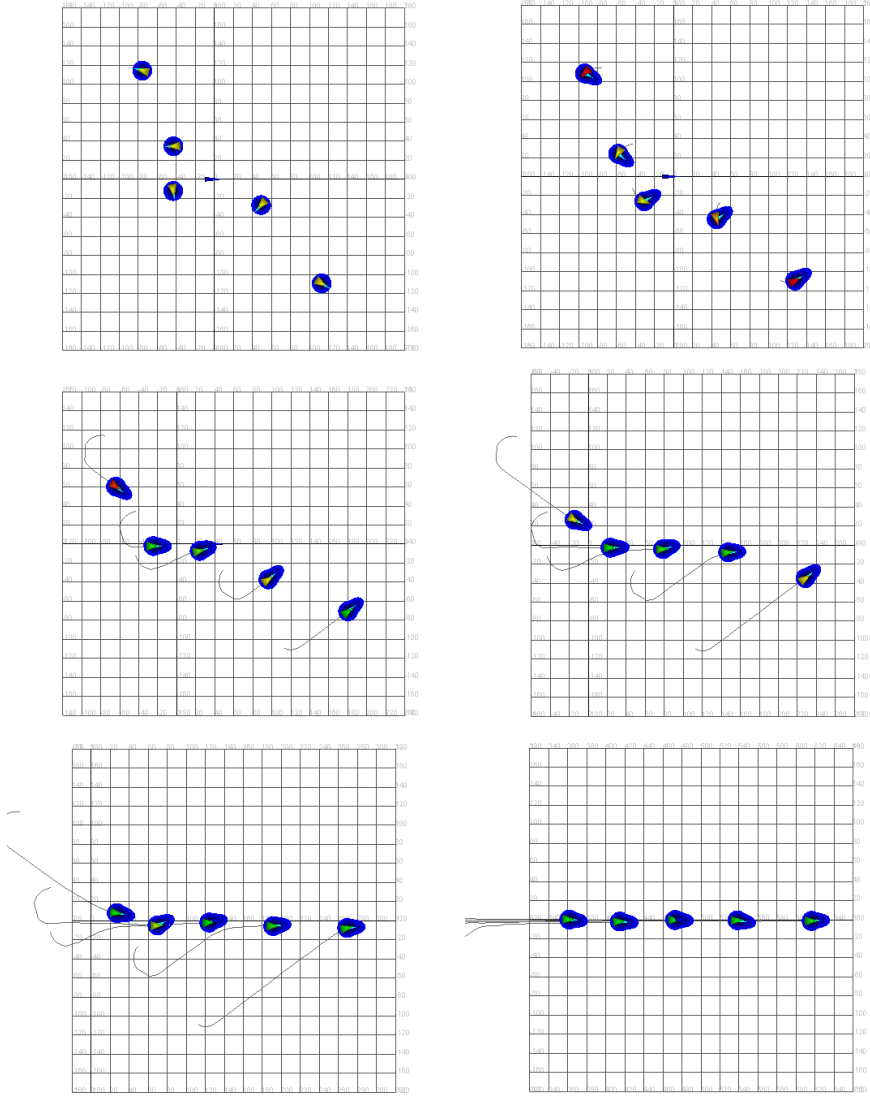


Figure 5.28: 5 Robots arranging a line formation

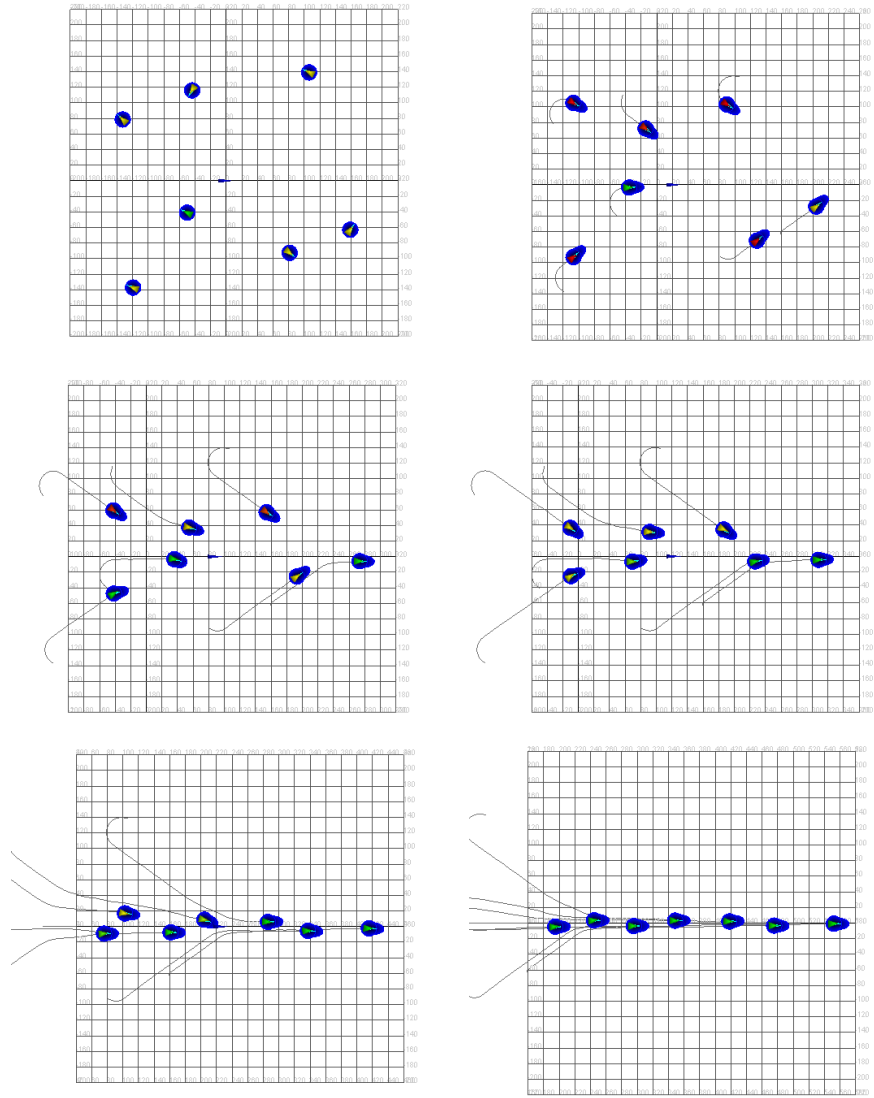


Figure 5.29: 7 Robots arranging a line formation

Comparing the fourth and fifth frame in figure 5.30, in the fourth frame there are three robots bordering the formation while in the fifth frame only two remain. The insertion point of the robot at the front right side of the formation in the fifth frame is obvious, there is a gap in the formation on the fourth frame that has been occupied on the fifth frame. However the two remaining robots show quite different behaviours. The robot at the left of the formation on the fourth frame is, in the fifth frame, still between the same two robots as in the fourth frame. While the robot at the middle on the right side of the formation at the fourth frame is, in the fifth frame, near the end of the formation.

Again the different behaviour of those two robots belongs to the action introduced in table 5.12. One robot, the right one slows down waiting for a possible gap in the formation behind him, the other robot, the one at the left of the formation keeps its speed and "pushes" towards the formation waiting for a gap to be built by the robots already in the formation.

The last frame, shows the final stable formation achieved by the 11 robots inside it.

In figure 5.31, with 20 robots to build up the formation, the robot-robot interactions keep growing: two, three or more robots can be observed to be involved, as they all aim for the limited area that represent the inside of the formation. However, given enough time, all the interactions are solved. It is interesting to consider that when formations with high amounts of robots are built, the direct influence of far robots is barely appreciable. Not only because the robot field decreases with the square of the distance, but when one robot "pushes" toward the formation, as the one commented in figure 5.30, this "pressure" has to travel from robot to robot along the connected robots within the formation. So, as the number of involved robots grows it takes longer and it is more ineffective for a robot to "push" its way into the formation. But, anyway, as it can be seen in the last frame of figure 5.31 the formation eventually reaches an stable configuration.

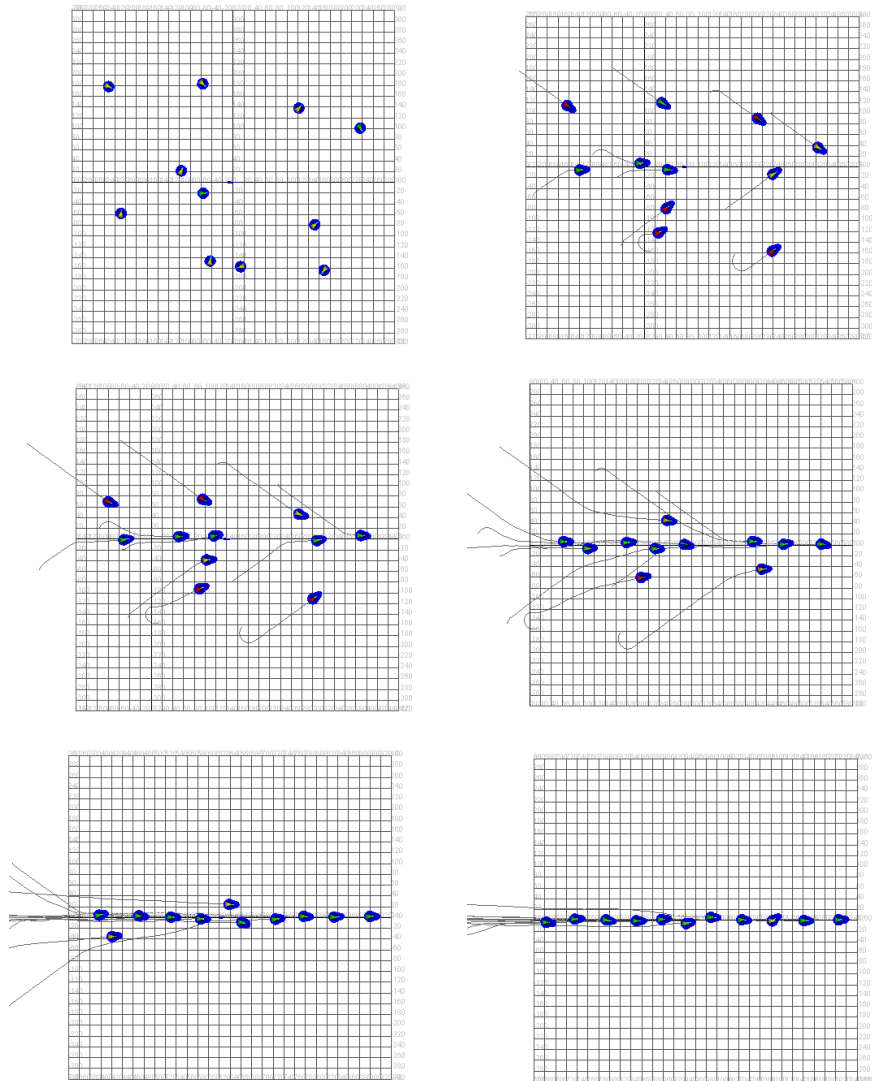


Figure 5.30: 11 Robots arranging a line formation

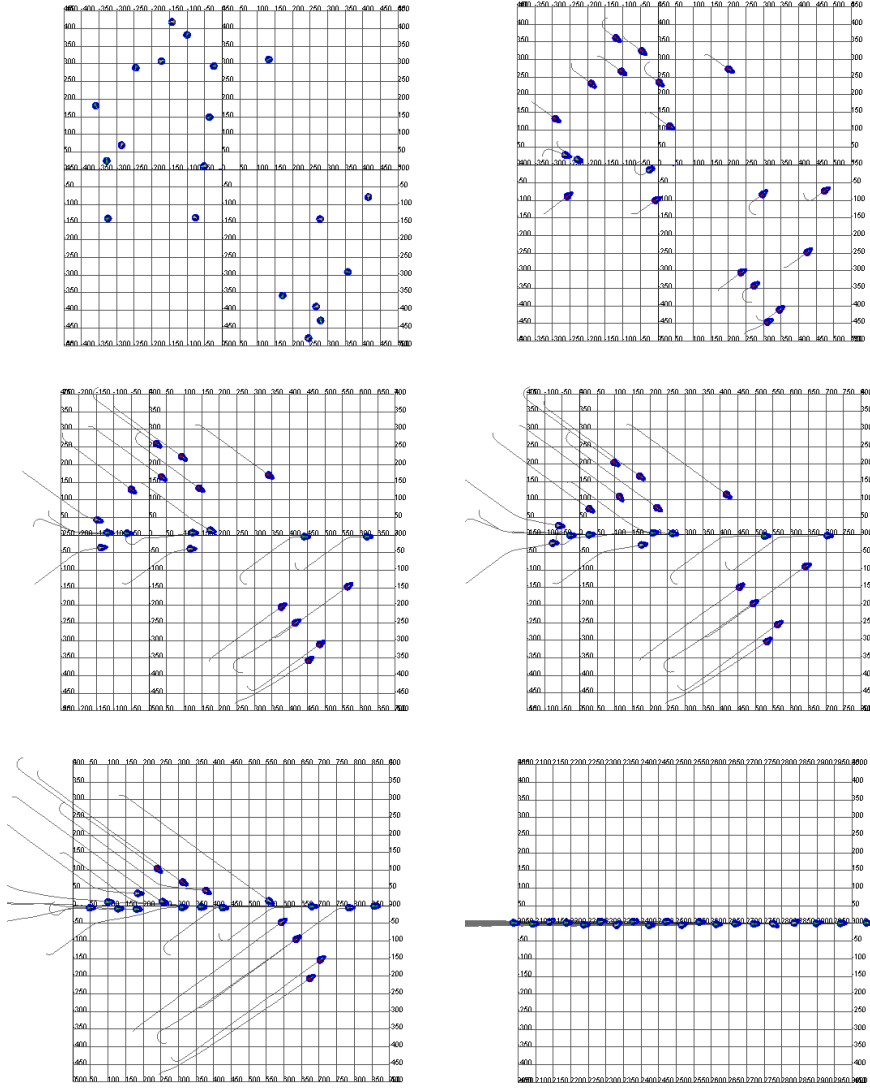


Figure 5.31: 20 Robots arranging a line formation

Column formation Along figures 5.32 to 5.35 a sample of simulations achieving of a column formation by groups of 5, 7, 11 and 20 robot is shown.

In figure 5.32, 5 robots in column formation, we can see the first of the column formation simulations.

At the first frame, the initial positions of the robots are the same initial positions that were used in figure 5.28, this is partly convenience and partly to show that different formations do not need specific initializations to be reached.

The second frame again shows the initial turning manoeuvre so the non-holonomous robots can align themselves with the general leader/formation direction. Notice that this initial turn differs from the one observed in the second frame at figure 5.28, this is to be expected since the formation field, also using a linear trace, has a different orientation relative to the formation moving direction.

Frames three and four in figure 5.32 shows how the robots align with the formation and leading directions. As they approaches the formation they have to keep distance within themselves and a few slight avoidance manoeuvres can be seen on those frames. It is more noticeable the behaviour of the robot at the bottom right along the frames. There, the robot aligns itself with the leading direction, but the travelling distance of this robot, when compared along the frames with the travelling distances of the other robots, is quite small; it barely moves. This is caused by its initial position, it is initially placed ahead of the formation, so following the actions stated in table 5.11 it aligns with the leading direction and brakes, waiting for the formation to reach its position, moment at which it begins to move together with the group as can be appreciated in the last two frames.

At the final frame on figure 5.32, when the robots have reached the stable formation it can be seen that one robot, the topmost along the figures, is not close to the other four. This is not a problem of the method, is just that nowhere in the stated situations there is an action to move the robots closer when the formation is achieved. This have been leaved as it is because a compact formation is only meaningful when the formations are open at the sides, as this one, but one of the formations to be tested is a circular one which will need a different action, to be equidistant to other robots, to "look nice". Those actions can be added at any time, but for the homogeneity of the tests they have not been included along the test simulations.

When two robots aim for the same position in the column formation the standard collision avoidance scheme is used and no other especial action to avoid symmetries needs to be taken, since the manoeuvring to avoid each other will not keep them out of the formation. Several of those situations can be appreciated in figure 5.33 where the robots just avoid the collision direction and keep moving towards the formation.

In the case of the column formation, the actions stated in table 5.12 to break the symmetry of two robots aiming for the same position inside the formation will never be applied since the situation evaluation will always be low. The action in 5.12 covers a special case due to a special symmetry situation that only appears when the formation trace is aligned with the formation moving direction.

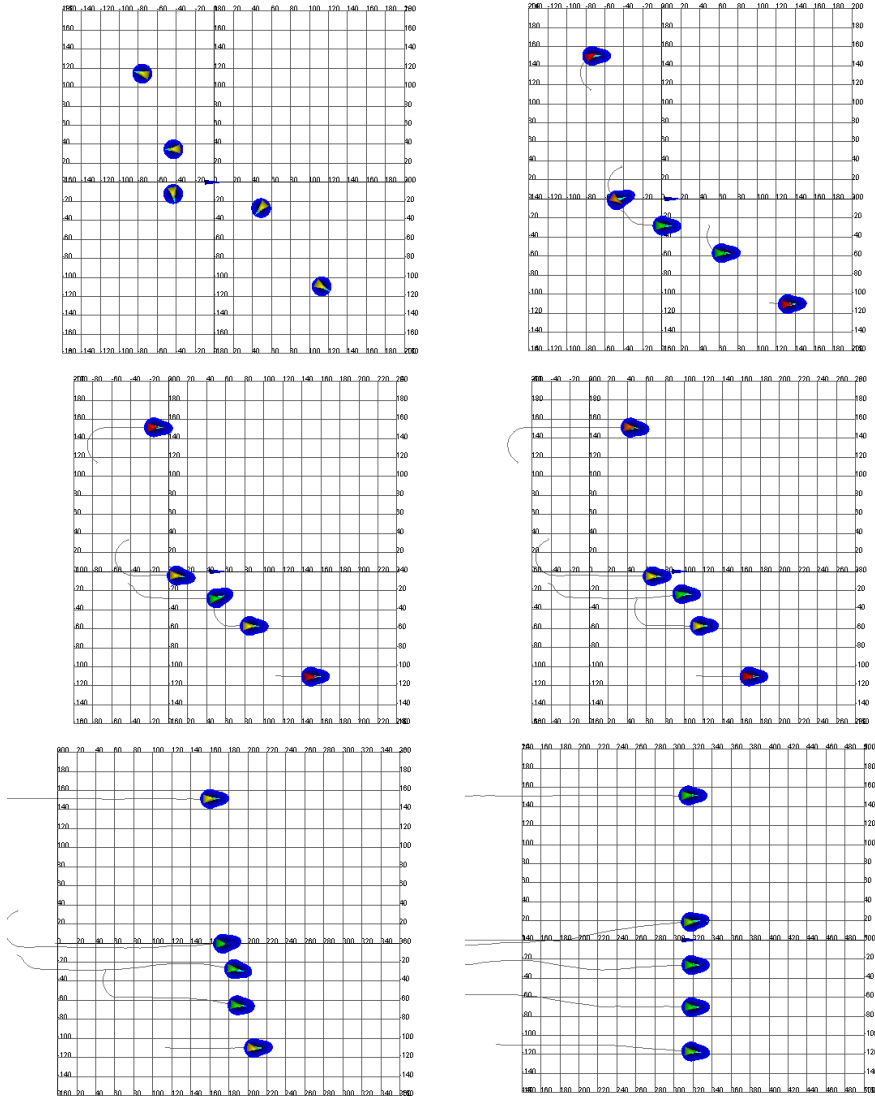


Figure 5.32: 5 Robots arranging a column formation

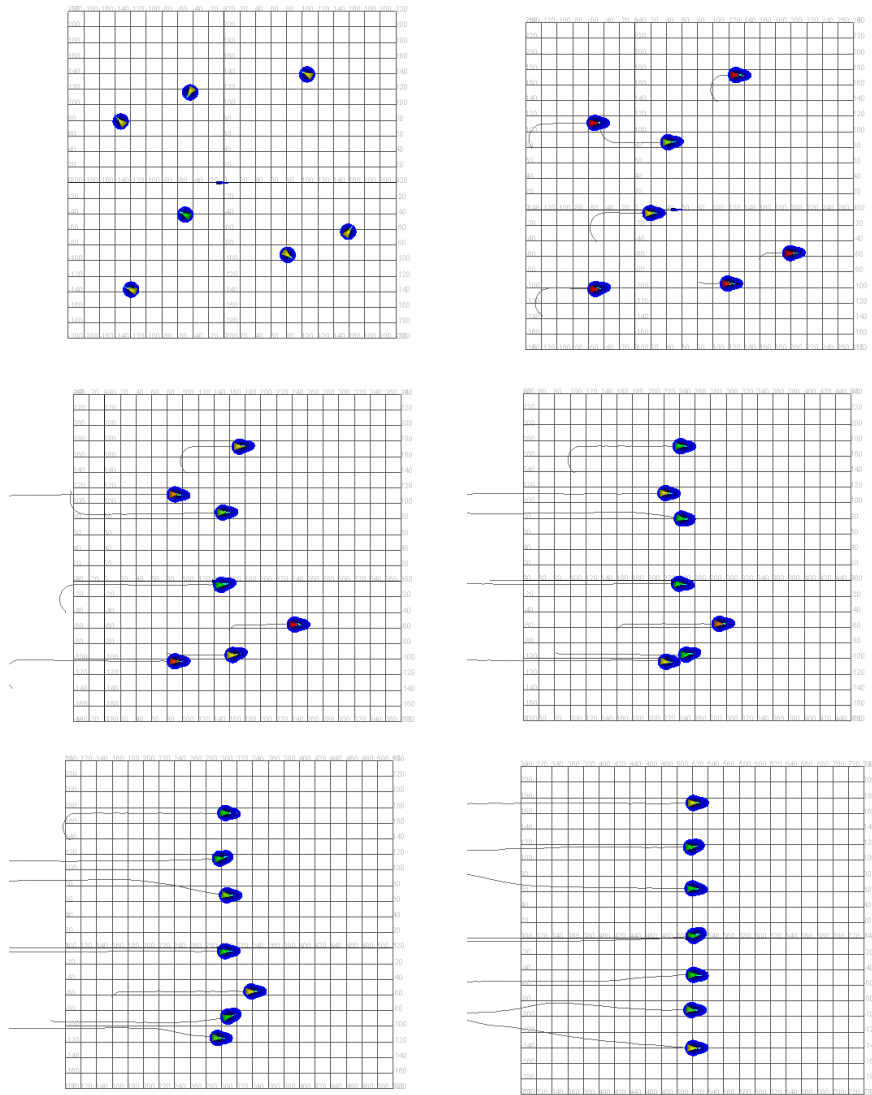


Figure 5.33: 7 Robots arranging a column formation

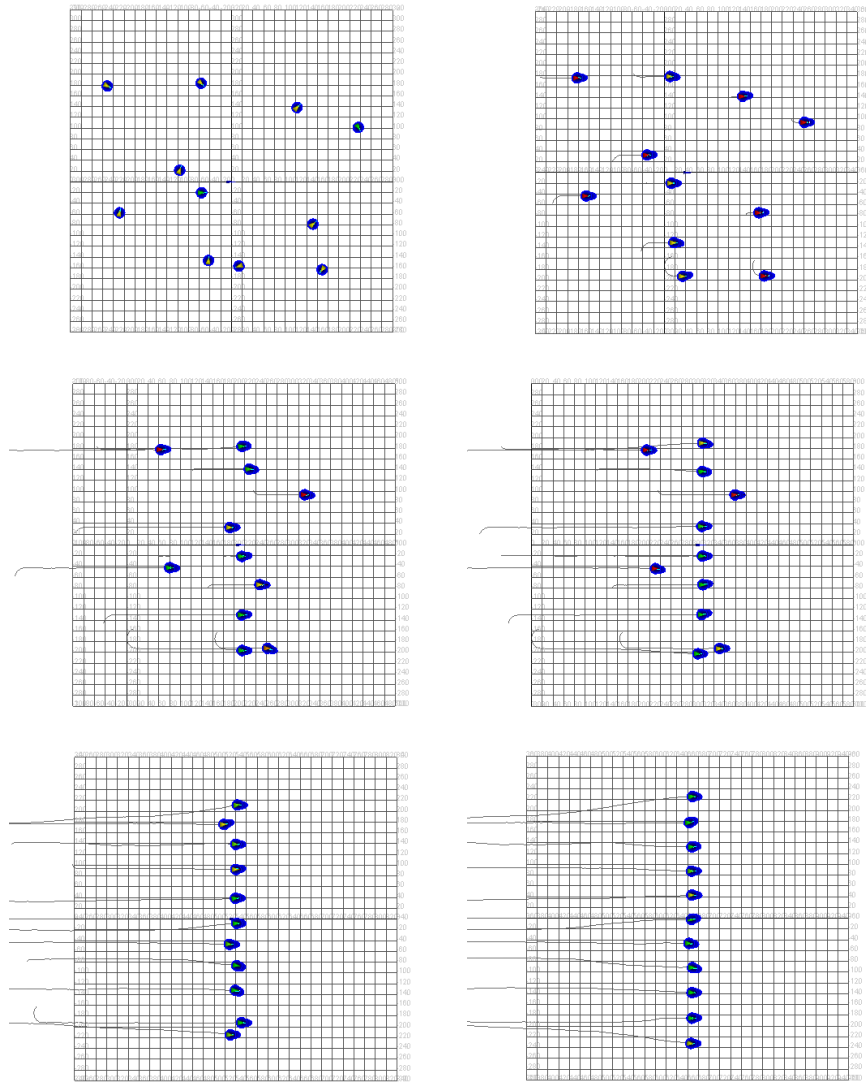


Figure 5.34: 11 Robots arranging a column formation

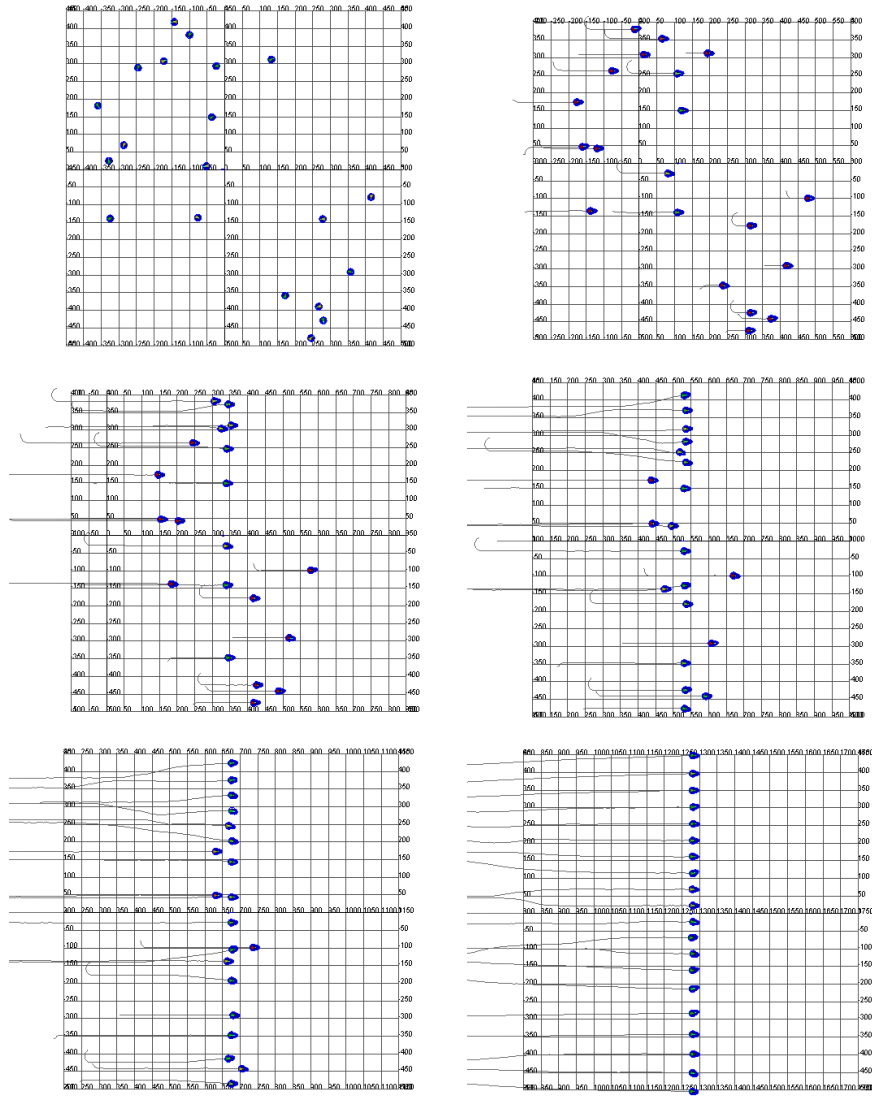


Figure 5.35: 20 Robots arranging a column formation

Circle formation Along figures 5.36 to 5.39 a sample of the simulations for 5, 7, 11 and 20 robots achieving of a circle formation is shown.

The main peculiarity of the circular formation is that each robot senses a different formation field direction relative to the leading direction, so assumptions on the field shape should be taken carefully.

At the second frame on figure 5.36, the initial manoeuvre evolves as in the previous formation shapes, only that here not all the robots align themselves equally, since for each of them, the formation field presents a different direction relative to the leading one.

The fourth and fifth frame show the formation nearly built. In the actions established at table 5.14 there is an Elect action to drive the robots away from each other once in formation, until they are on the far side of the manoeuvring distance. This distance was set at $6 \cdot r_{Lim}$ but the formation circle radius is set to accommodate the robots with a distance of $5 \cdot r_{Lim}$ between them. Since there is less space inside the circle than the established distance between robots there is always a little pressure inside the formation. This was intentionally done since it results into more compact final formations and barely affects the system evolution. In frames four to six it can be seen how the robots move, once inside the formation, to accommodate an nearly equidistant final state which minimizes the imposed "pressure".

As the number of robots inside the formation grows, more robot-robot interactions can be observed. In a circular formation all possible relationships between robot field direction, formation direction and leading direction can happen and robot-robot interactions within robots affected by different field directions will happen.

It should be noted that the circular formation has been a notable source of information along the process of building situation-action sets, because of the multiple interactions that take place in it, highlighting problems that will be difficult to point using straight line formations. In those cases a stall on the robot movement towards the formation can be observed as a result of a conflict between different situation-action statements.

Also it should be pointed that, when the number of robots grows large enough and the circle radius is long enough, the readings of the field sensors are barely different from those observed when the robot approaches a linear formation. This is to be expected due to the inverse of square relationship of the fields with distance, however it supports the idea that scenarios with even more robots in them will behave mainly as the studied ones since the farther robots and farther parts of the formation field will not affect substantially the sensor readings of the robots and they will act locally as if they were in a smaller formation.

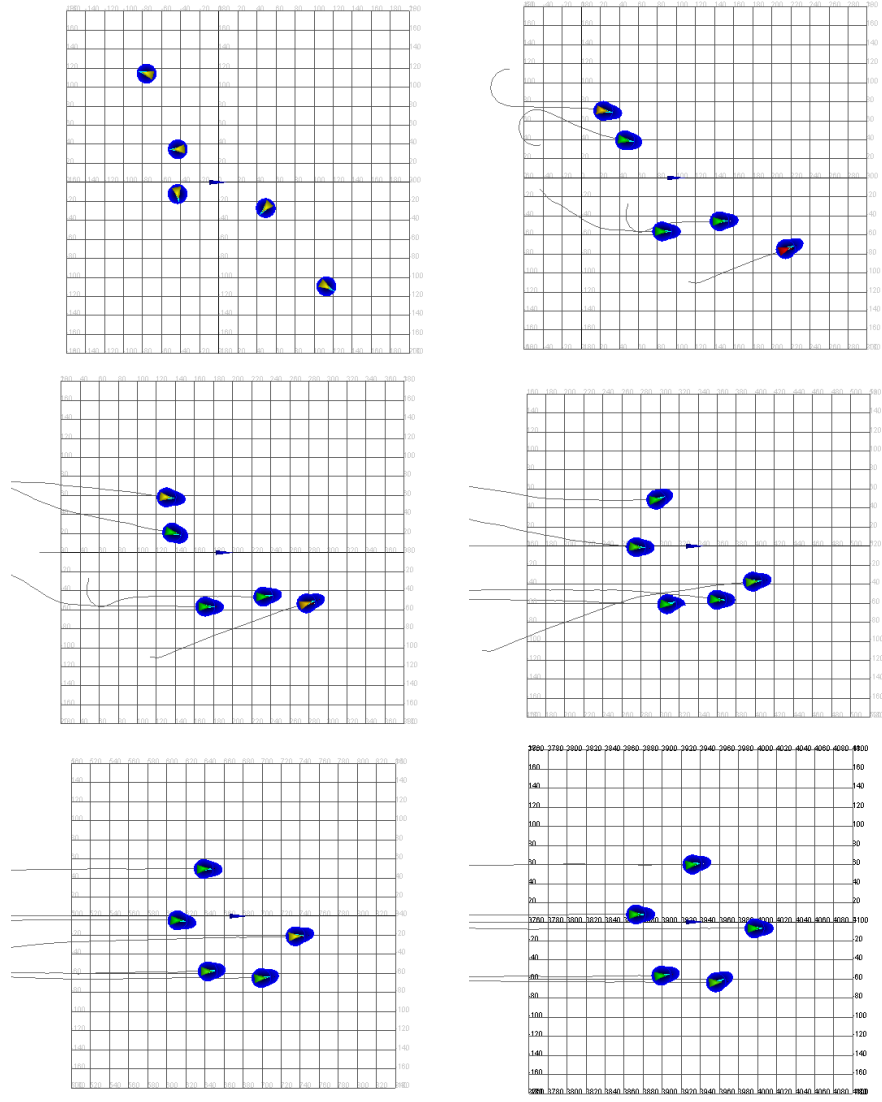


Figure 5.36: 5 Robots arranging a circle formation

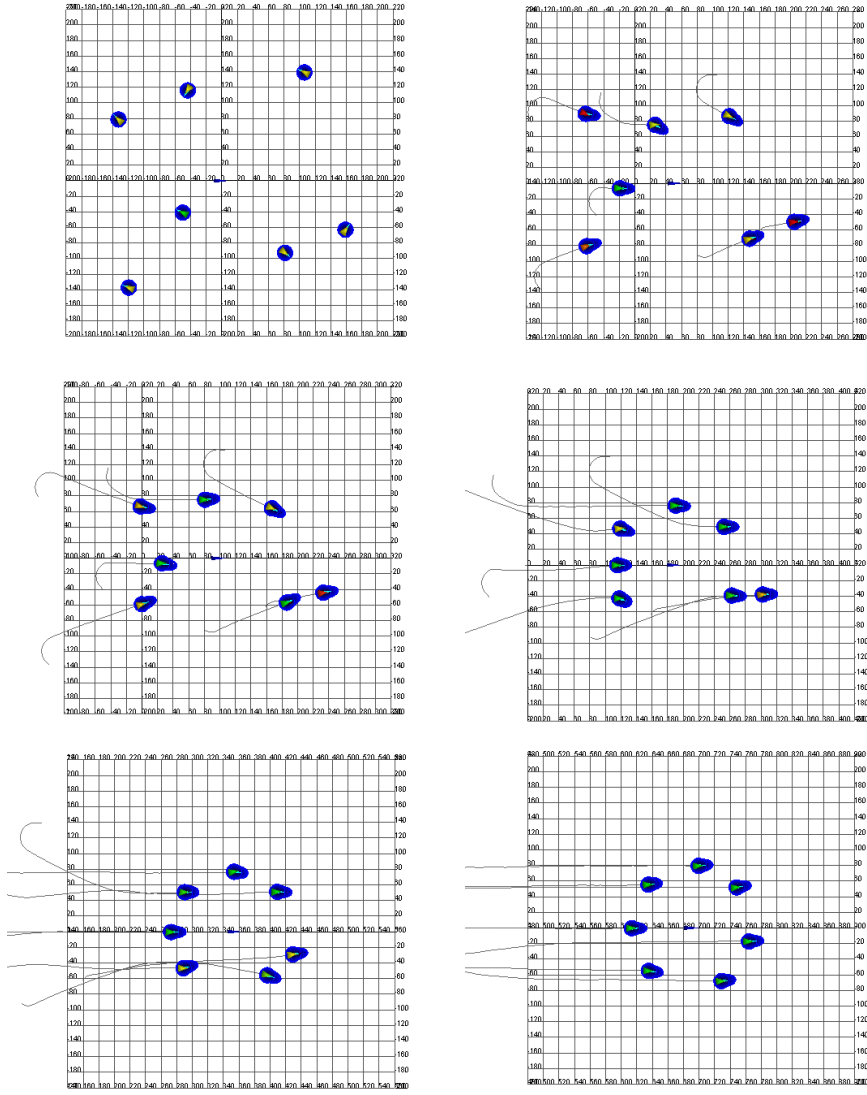


Figure 5.37: 7 Robots arranging a circle formation

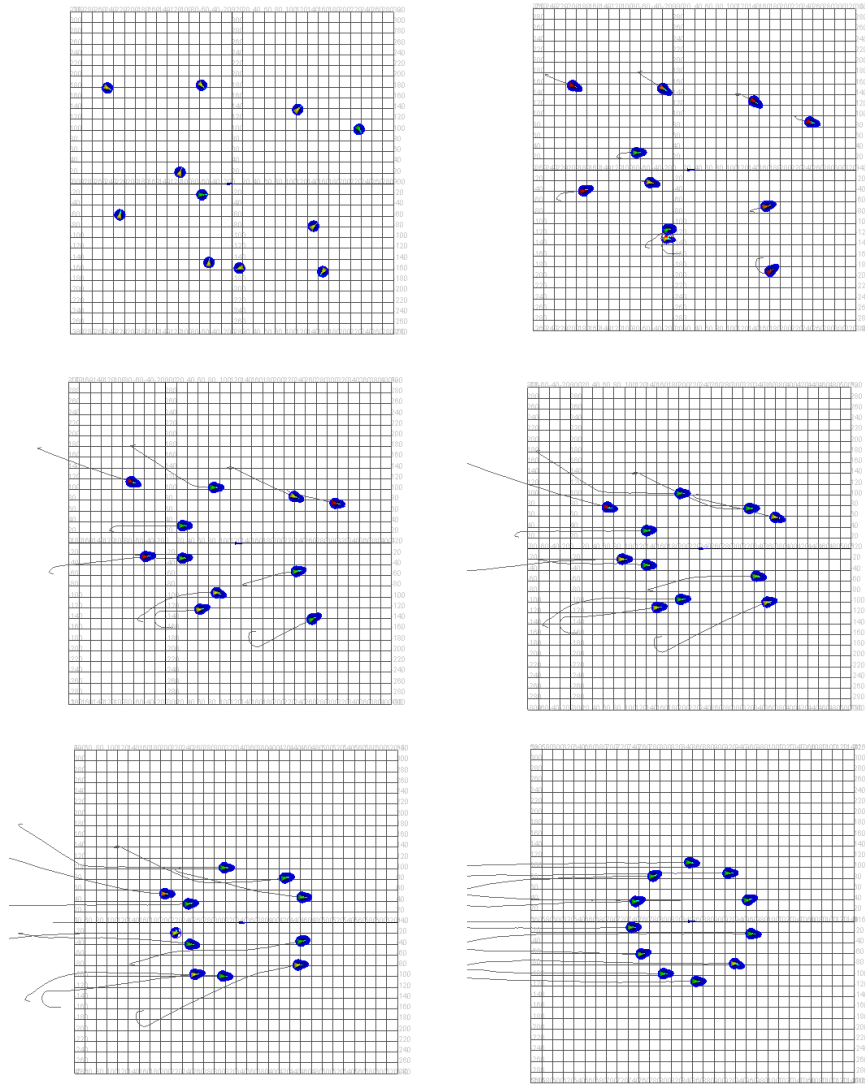


Figure 5.38: 11 Robots arranging a circle formation

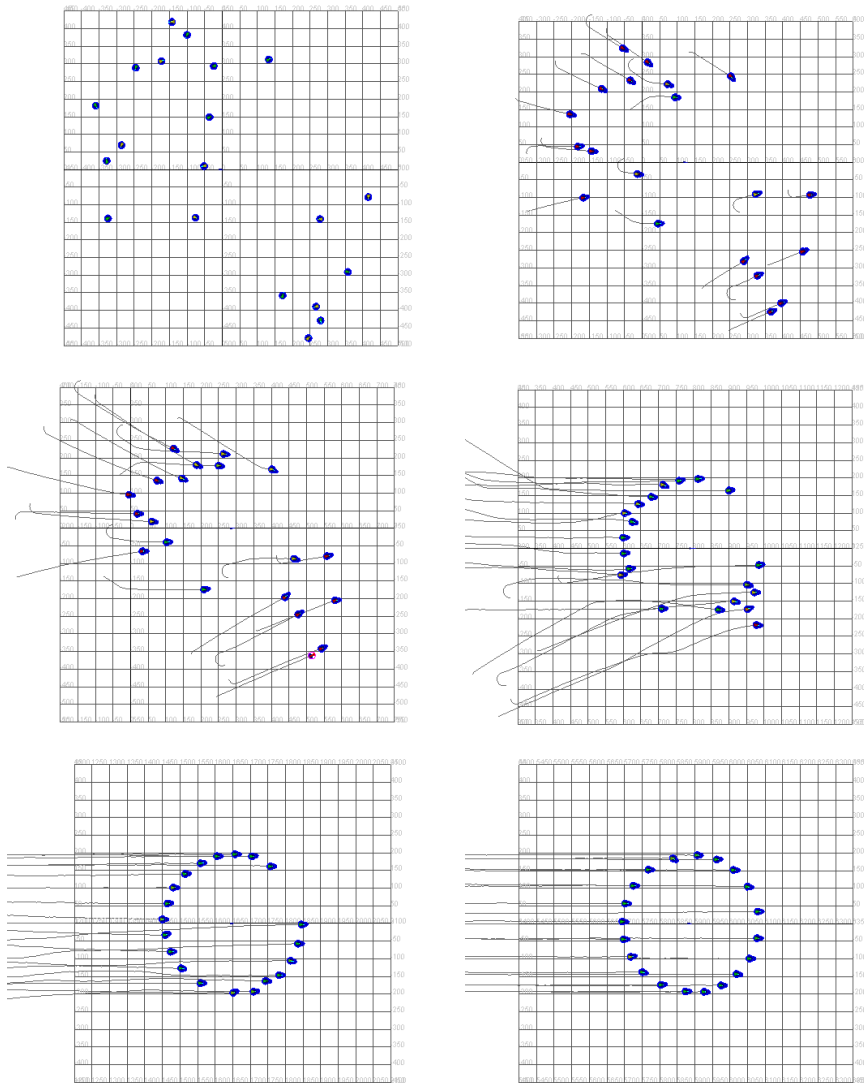


Figure 5.39: 20 Robots arranging a circle formation

Wedge formation Along figures 5.40 to 5.43 a sample of the simulations for 5, 7, 11 and 20 robots achieving a wedge formation is shown.

At figure 5.40, the initial manoeuvres for the wedge formation are essentially the same that in the previous cases: the robots turn to reach the approximation orientation.

One difference to empathise is the behaviour of those robots ahead of the formation which suffer a discontinuity in the sensed formation field when they cross ahead of the formation vertex –the formation field expression for the wedge formation has been built using this discontinuity to show its influence in the robots–. In the third frame the robot ahead of the formation can be seen moving toward the vertex; in the fourth frame it is still ahead of the formation but the waiting process has been accompanied by slight oscillations around the line ahead of the vertex. This behaviour is due to the angle between the formation field and the leading direction, which moves the robot toward the vertex, and the actions expressed in $S_{3.1}$ at table 5.11, which impels the robot to wait aligned with the leading direction. The easiest way to avoid this, if needed, is just to provide a field with a continuous gradient around the vertex but it is interesting to observe that the discontinuity only induces a small oscillation and does not seriously affect the formation gathering

In the fifth and sixth frame, as the front robot gets inside the formation, the oscillation disappears due to the change of dominant situation.

In any other aspect the wedge formation evolves as the previous formations reaching again a stable formation.

In figure 5.43, 20 robots in wedge formation, at the fifth frame, the two robots behind the formation structure at the upper part of the figure, are being partially affected by the action in table 5.12 intended to solve the symmetry on formations aligned along the moving direction. When this situation affects partially to formations, like this one, its effect will be that those robots at the right side of the formation trace will skirt the formation until a empty spot in the formation is found, however those robots at the left side of the formation trace will wait for the robots in the formation to open a gap for them.

The final stable formation is reached on the sixth frame. The observable gap is due to the lack of a situation-action element to get all the robots together as was explained in the case of the column formation.

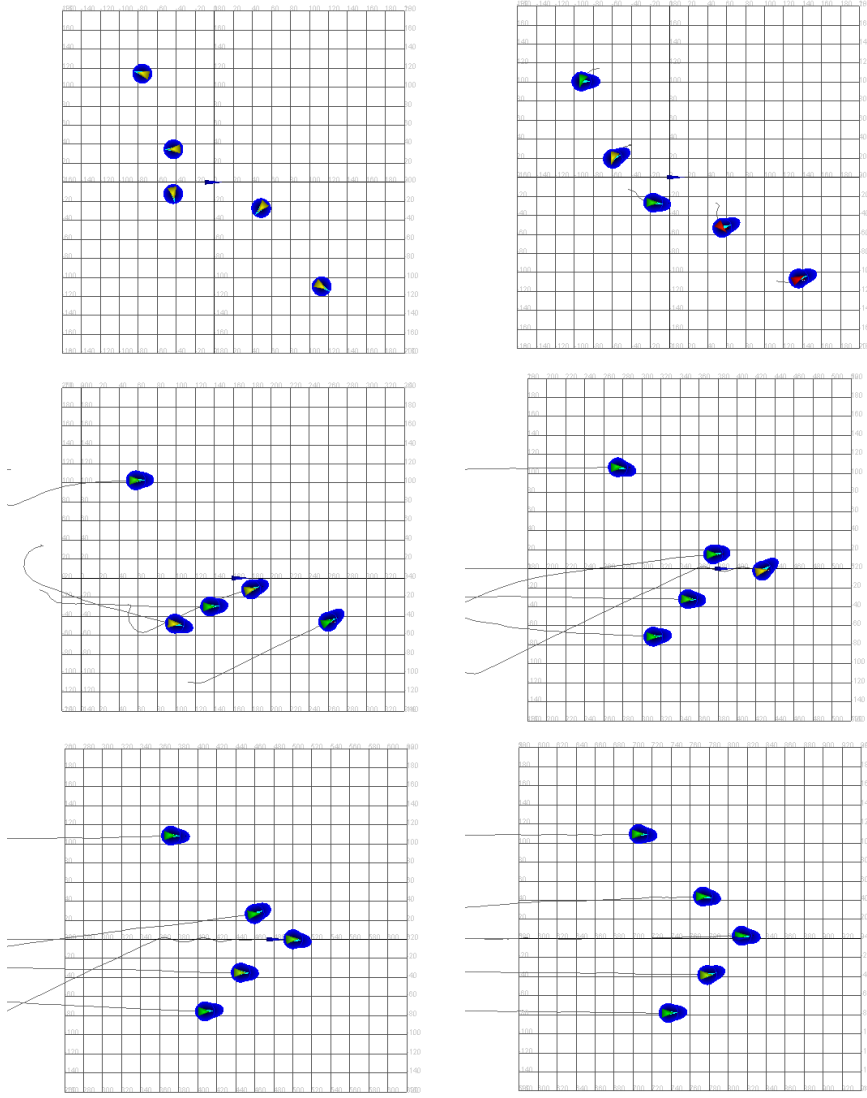


Figure 5.40: 5 Robots arranging a wedge formation

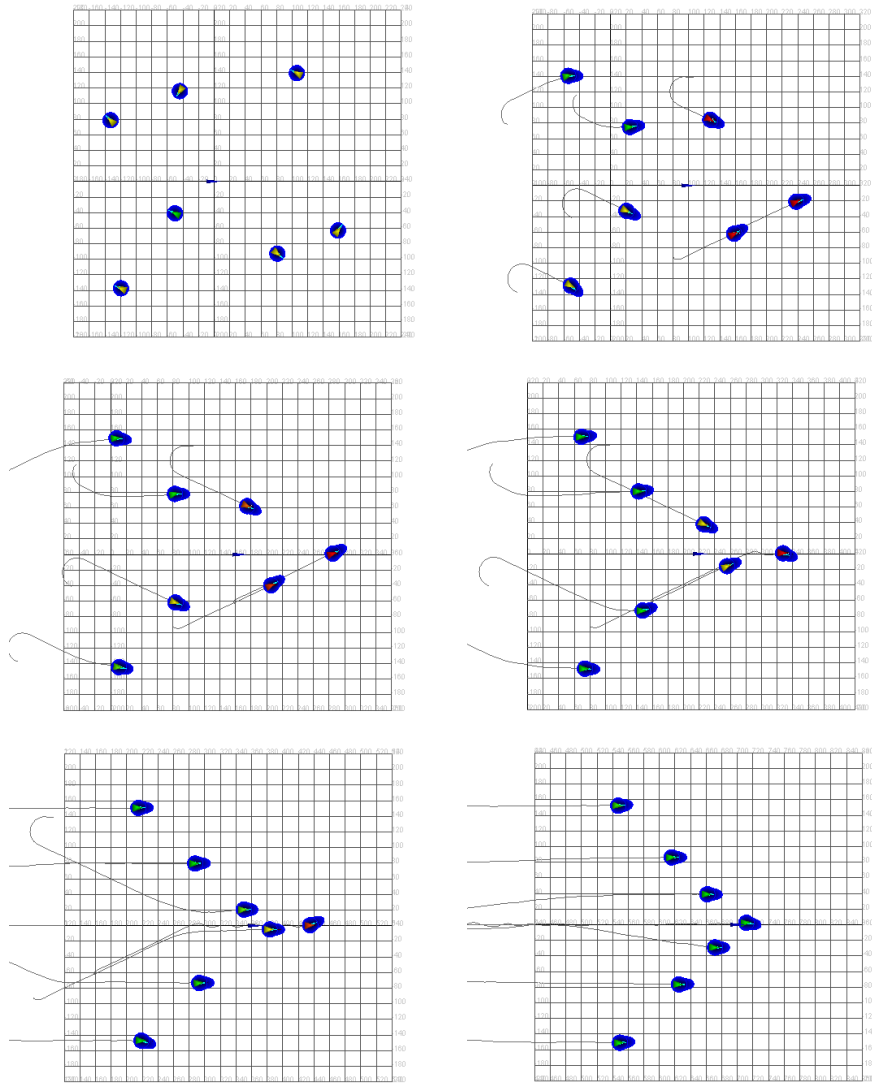


Figure 5.41: 7 Robots arranging a wedge formation

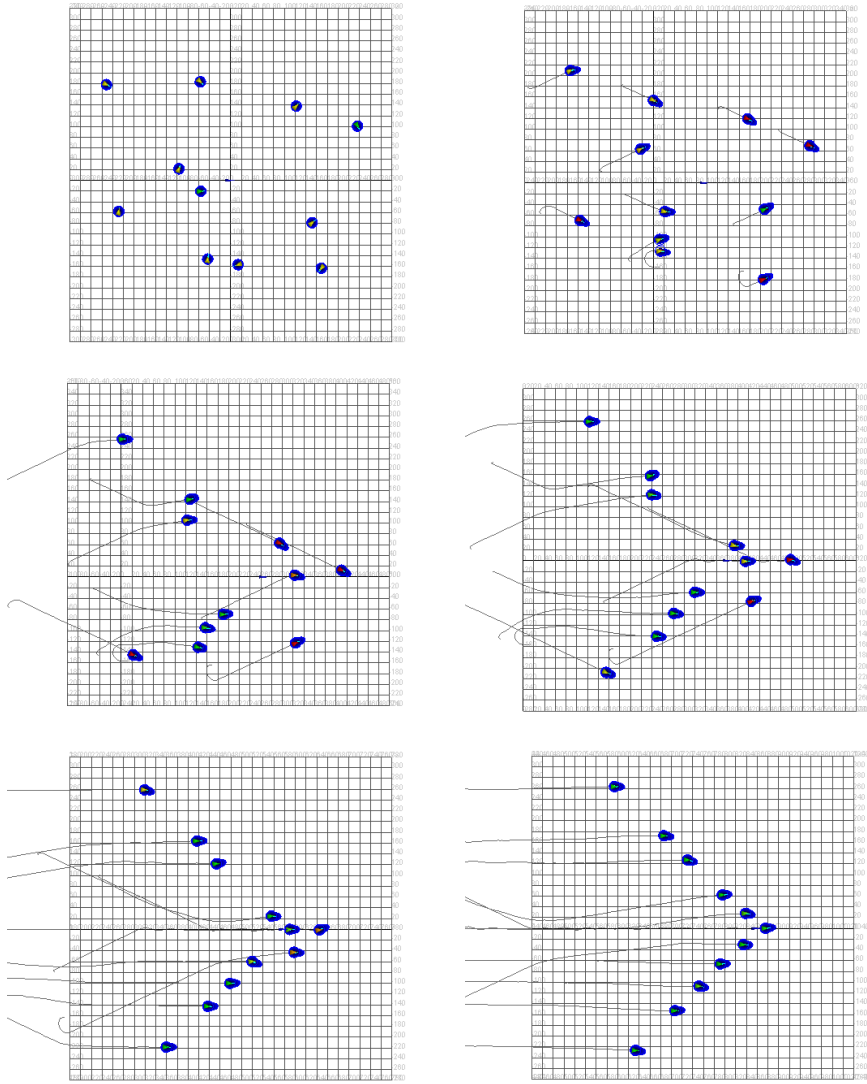


Figure 5.42: 11 Robots arranging a wedge formation

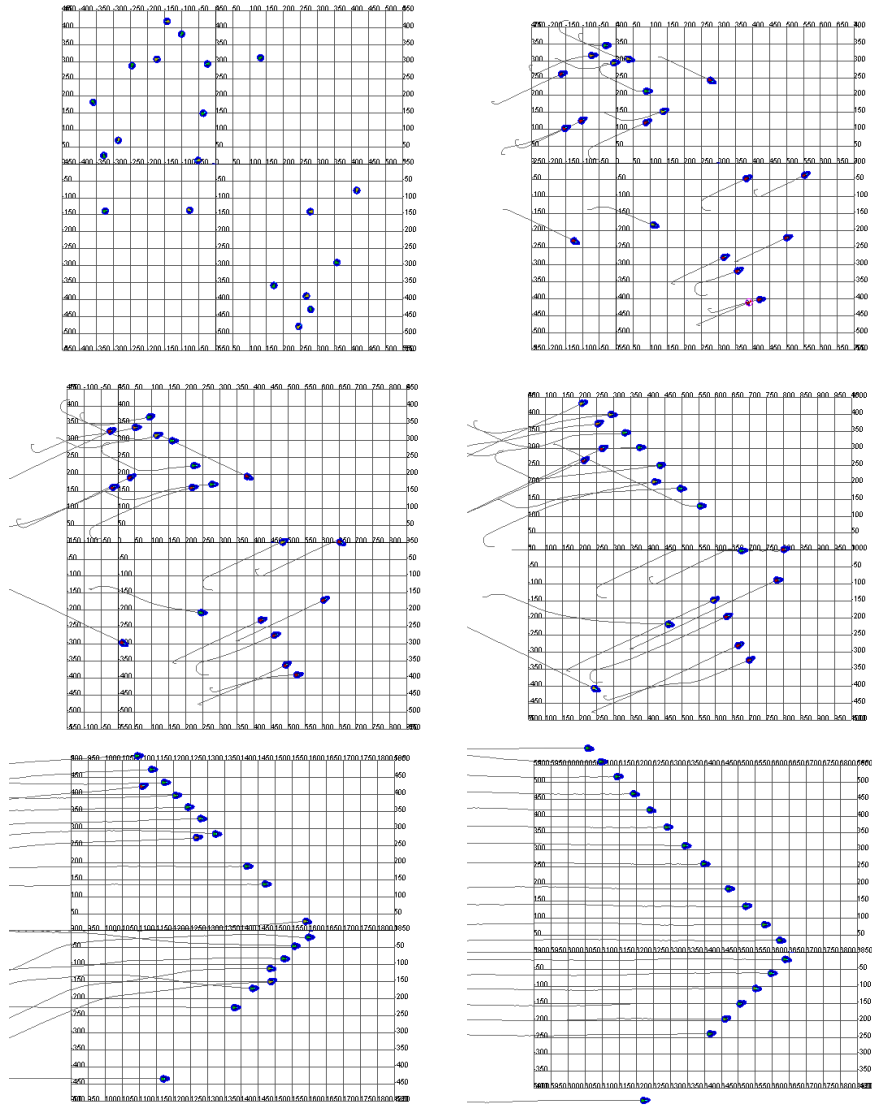


Figure 5.43: 20 Robots arranging a wedge formation

Statistical Analysis

The previous images provide an insight about how the formation gathering evolves along time.

A more general view of the results for the formation gathering process is obtained through systematic simulations done with random initial positions and orientations for the robots. To obtain relevant results about the formation gathering process, over 100 simulations have been done for each one of the four field shapes and for ten different formation sizes : 3, 4, 5, 6, 7, 9, 11, 15, 20 and 30 Robots.

The result is more than 4000 different scenarios and nearly 45000 robot simulated, which will provide enough statistical data about the process.

To extend the testing of the proposed method, these simulations are done twice, one with a perfect, noiseless, simulation and a second time injecting noise into the sensed distances used to build the fields, into the sensed own position and orientation of each robot and into the effective movement of each robot.

Distance to Formation The main target of the formation cases is to drive the robots toward the formation structure, so a good indicator for the achievement of the objective is the distance from the robots to the formation structure along time. While each scenario will be different it is not to be expected the same result from every one of them, however a similar tendency can be expected for those simulations that share the formation shape and/or the amount of robots in them.

The next figures, figures 5.44 to 5.47, show the mean distance of the robots to the formation along time within its standard deviation for the different formation shapes and amount of robots in them.

The simulation step and sampling time used to obtain these data is 0.1s. Each one of the curves shown below is built using the measurements of 101 simulations for each combination of number of robots and formation shape.

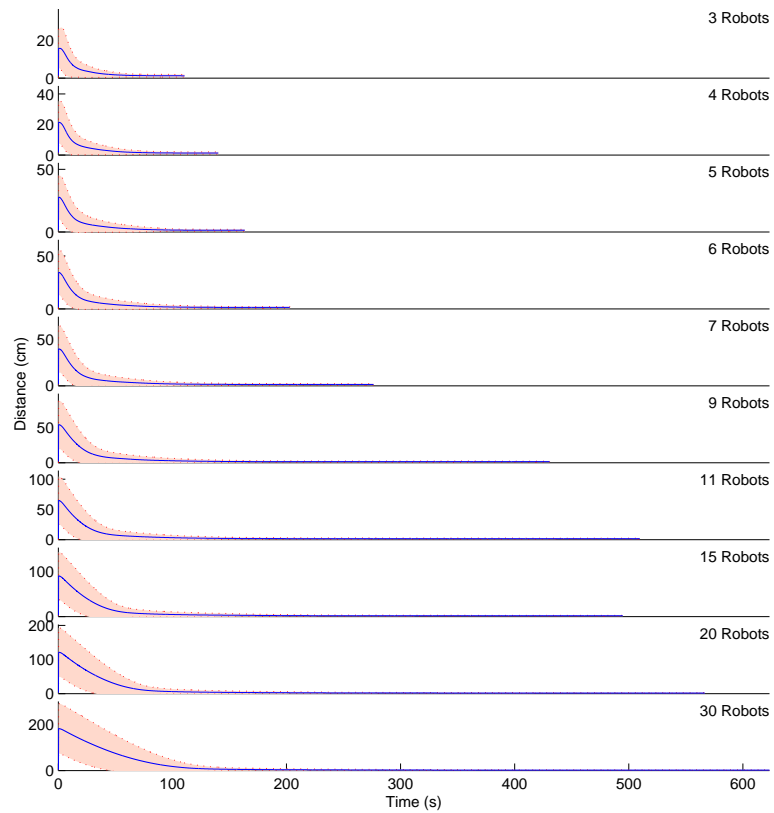


Figure 5.44: Line formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations

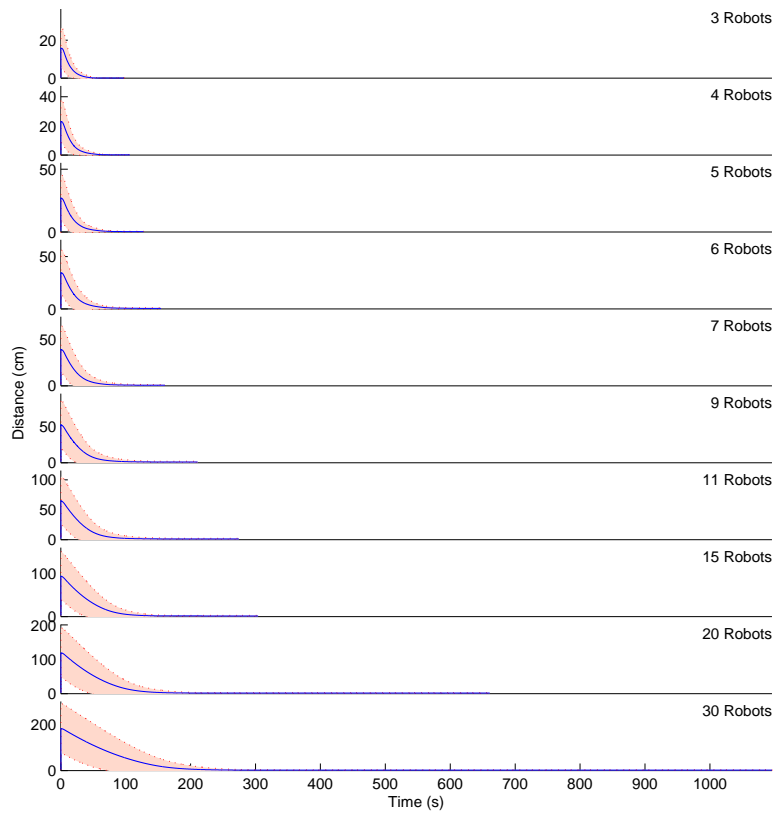


Figure 5.45: Column formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations

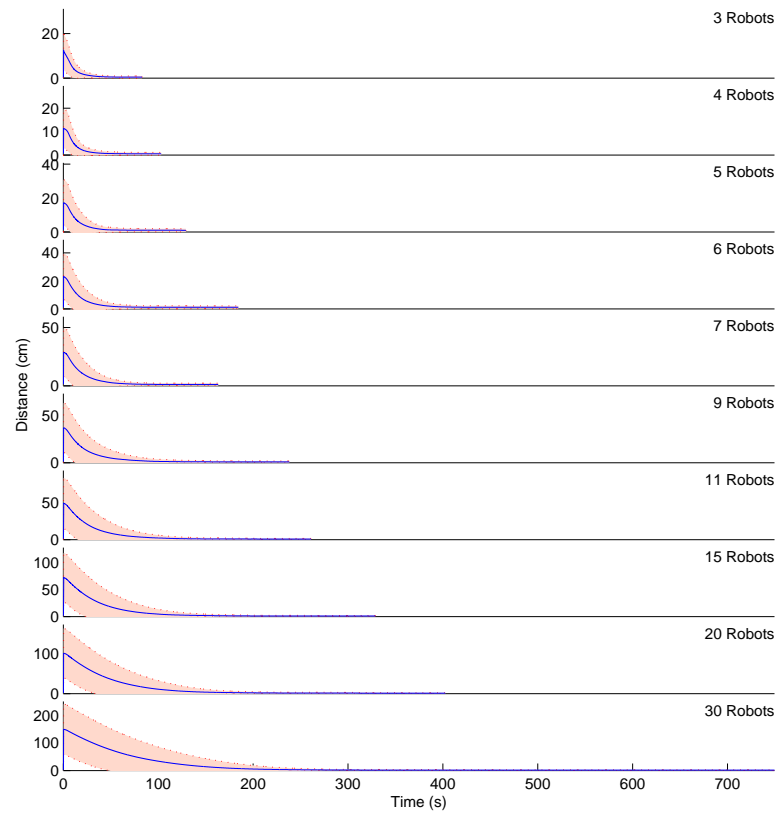


Figure 5.46: Circle formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations

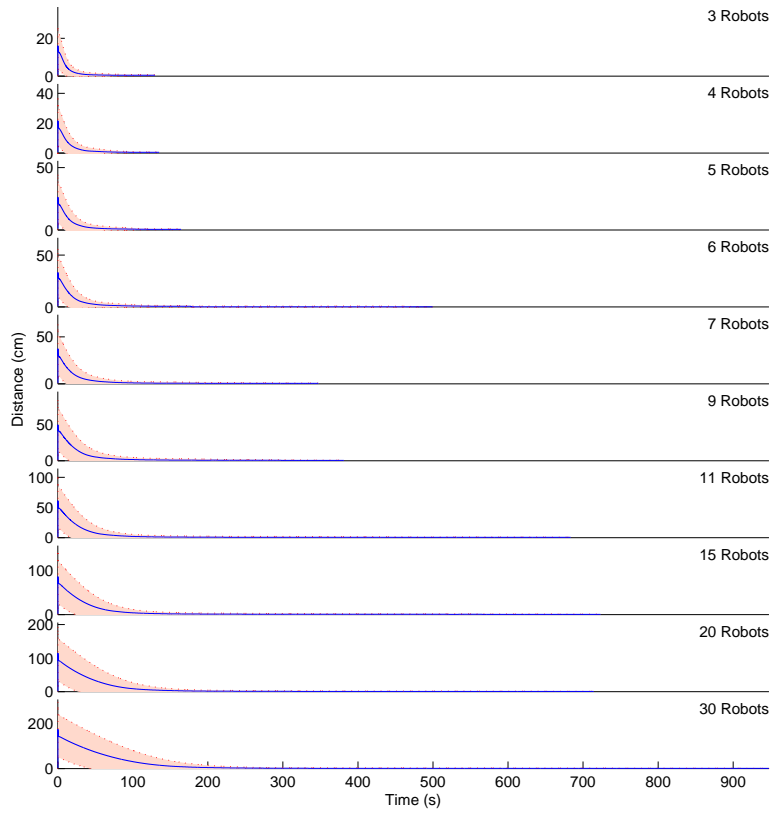


Figure 5.47: Wedge formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations

Distance To Formation Fit The results for the distance to the formation on all simulations shown in 5.44 to 5.47 show a clear similitude. To study this similitude, and in order to extract some numerical estimations about the distance to the formation along time, all the mean distance curves have been fit to the exponential decay function found in eq. 5.55.

$$d = a \cdot e^{-t/b} + c \quad (5.55)$$

At equation 5.55, the three parameters to adjust are a , b and c , while d represents the distance to the formation and t the time.

For the three parameters to adjust the first one, a , will reflect the level of initial dispersion of the robots from the formation line, –since the values for c are much lower than those from a –. This parameter is only of punctual interest since it only reflects a characteristic of the specific tests, not something related with the performance of the robots in their gathering. The linearity of this parameter is expected since the initial random position of the robots is done over a square with sides proportional to the number of robots.

Parameters, b and c reflect characteristics of the method and the gathering process.

The second parameter, b , establishes a time constant of the gathering process. This time constant reflects the rate at which the robots moves nearer to the formation. Knowing the dispersion of the robots and its time constant an estimate of the time that the robots will need to gather up can be obtained.

The third parameter, c , provides an estimation about the mean distance that the robots will keep to the formation when it stabilizes. This third parameter does not shows perfect zero due to the construction of actions associated with situation $S_{3,2}$ –when the robot is inside the formation–. These actions adjust the robot position proportionally to its distance to the formation trace, and only when this distance is above $r_{Lim}/10$ –due to the definition of 5.42–. When the value of $S_{3,2}$ is small enough the corrections are small enough to be absorbed by the oscillations of the formation structure positioning algorithm coupled with the corrections of the individual robots in the formations.

The results for these fittings are shown in figure 5.48 where a clear linear relationship between the two first fitting parameters and the number of robots in the formation can be seen, while the third fitting parameter remains in the same region, mainly independent on the number of robots.

Observing the results of the fitting for the different amount of robots within the formations and the different formation shapes, it is noticeable that the second fit parameter, b , shows a clear linearity with the amount of robots within the formation and how it also displays reasonable similar results for all the formation shapes. The observed linearity allows the estimation of the gathering rate for other scenarios with previously untested amount of robots and a rough estimation for untested formation shapes. From the obtained results of the second parameter b , it can be considered that the line formation and the column formation are the extreme cases for the gathering rate; other formation shapes gathering rates will probably fall between these two formations gathering rates.

Looking at the results of the third fitting parameter c it shows how the robots move towards the formation but they do not fit perfectly the formation trace

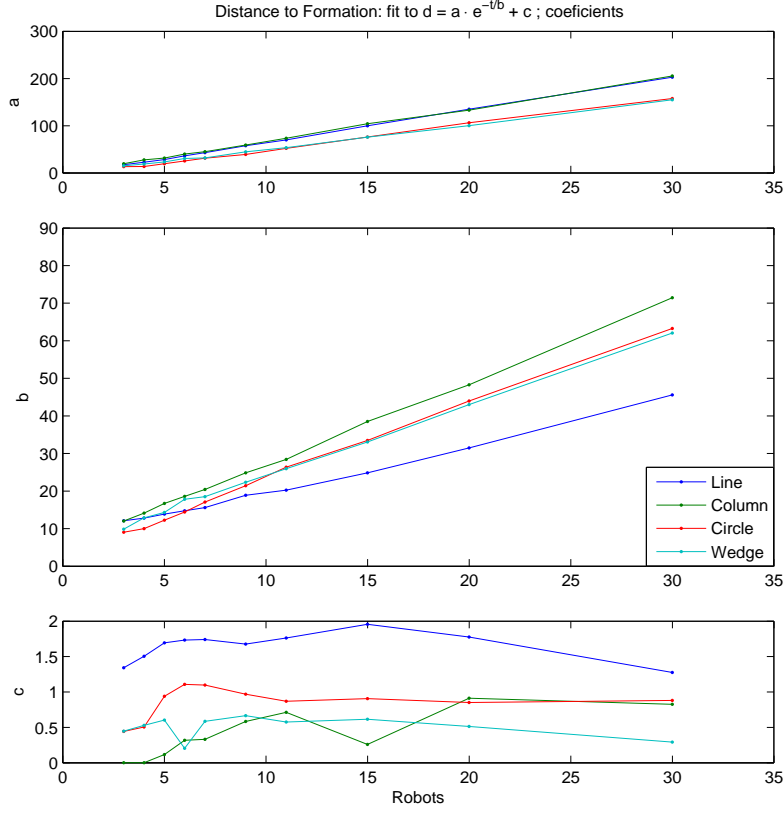


Figure 5.48: Coefficients from the fitting of the distance to formation along simulation time to a exponential function for different number of robots in the formation

–the fit value remains above 0–, however the remnant distance, not greater than $0.2 \cdot r_{Lim}$, is small enough to consider the formation shaped, since the actions guiding the robots consider the robot completely inside the formation –an do not act– when the distance to the formation is below $r_{Lim}/10$.

As with the mean distance, the standard deviation of the distance between robots and formation has been also fitted to the same exponential decay function. The results for the fitting parameters are shown in figure 5.49 and again the fitting parameters show a clear linearity with the number of robots building up the formation.

The standard deviation on the distance between the robots and formation trace is related with the amount of dispersion of the robots from the formation shape.

The first parameter, a , again shows the initial dispersion of the robots, since they were randomly dispersed it is expected to be high, and this can be seen in

figures 5.44 to 5.47 where the initial deviation is, as expected, in the order of the mean distance.

The second parameter of the fitting is again the most relevant one, the time constant of the exponential decay on the robot dispersion. The fitting of dispersion on the distance to the formation into an exponential decay function represents that the gathering process is taking place. Not only the robots move in average nearer to the formation trace but the group gets tighter as time passes.

The third parameter, c , on the fitting of the deviation on the distance to formation, represent the remnant dispersion of the robots when the formation is stable. Being a non-zero mean distance to the formation at the final stable state is expected that the dispersion was also non-zero due to the small oscillations of the robots and the formation trace and the lack of correction actions when the distance to the formation is below $r_{Lim}/10$.

The previous fittings show us that both: the mean distance of the robots to the formation and the dispersion of the robots from the shape do decay in an exponential way. Considering this we can expect that the robots in a scenario will gather around the provided shape –the mean distance decays– and also that the shape will increase its definition along time –the dispersion around the mean distance also decays– until it reaches a limit value –both values do not decay to zero but to a lower limit value–. This behaviour can be observed in the provided examples, and can be expected in other cases and formation shapes also, since all the tested shapes and number of robots have fallen under the same fitting schemes.

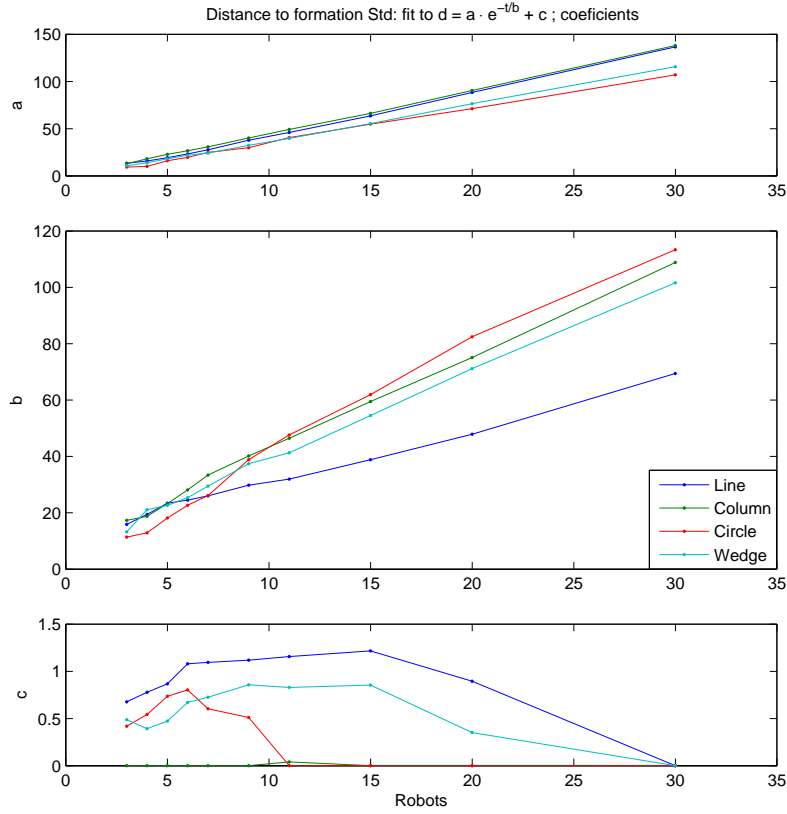


Figure 5.49: Coefficients from the fitting of the standard deviation on distance to formation along simulation time to a exponential function for different number of robots in the formation

Arrangement Time The second significant value that is going to be used to study the behaviour of the method is the time needed by a group of robots to reach the formation state. The analysis of the distance to the formation trace shows the individual robot navigation capabilities. Looking at the arrangement time we can observe the group coordination and how the situation-actions sets of the different robots interact, revealing a coordinated structure.

As can be deduced from the formation situations, there is not really a "final" robot position within the formations; there are not specific points for each robot to reach and there is not a point where the formation configuration freezes.

However when each robot in the formation reaches a distance to the formation trace low enough, with every other robot at a distance higher than the collision distance, and when this situation is kept for a given period of time, it can be considered that the formation has reached a final stable configuration –it has been arranged–, since only external influences can disrupt such situation.

The formation arrangement time is established, for testing purposes, when every robot in the formation is at a distance to the formation trace below $r_{Lim}/2$, the distance to any other robot is higher than the collision distance, and this situation lasts 50 seconds –500 control iterations–.

Histograms for the arrangement time for the different simulated scenarios are shown in figures 5.50 to 5.53, this establishes a basic idea of how long it takes to build up the formation in the different combinations of formations and number of robots.

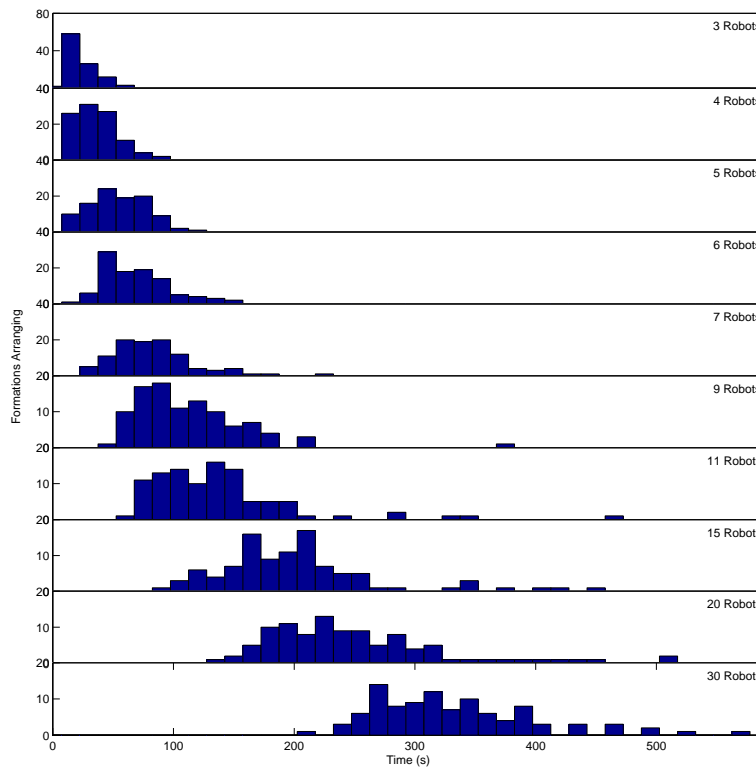


Figure 5.50: Line formation, Formation arrangement time histograms for 101 simulations

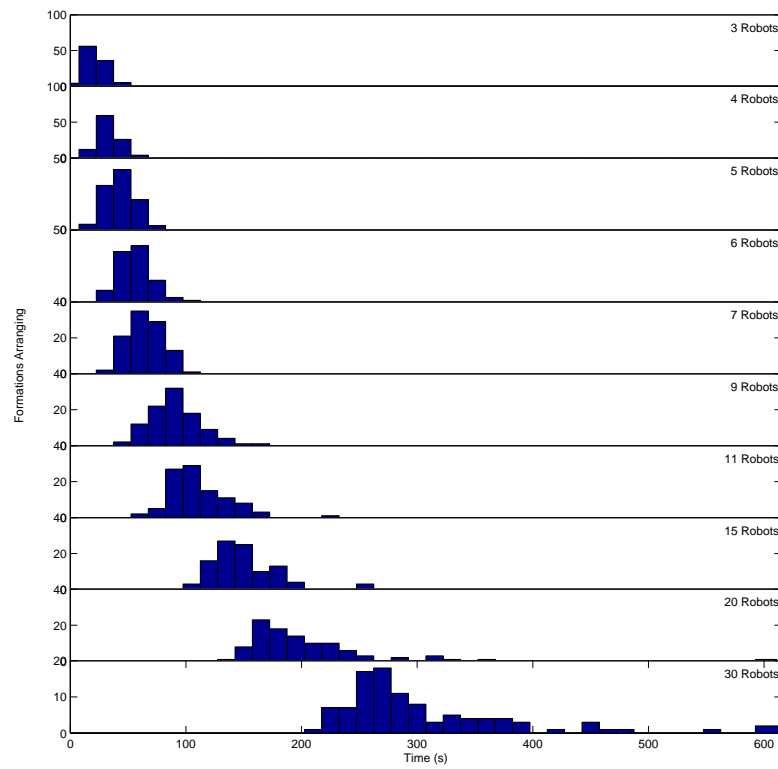


Figure 5.51: Column formation, Formation arrangement time histograms for 101 simulations

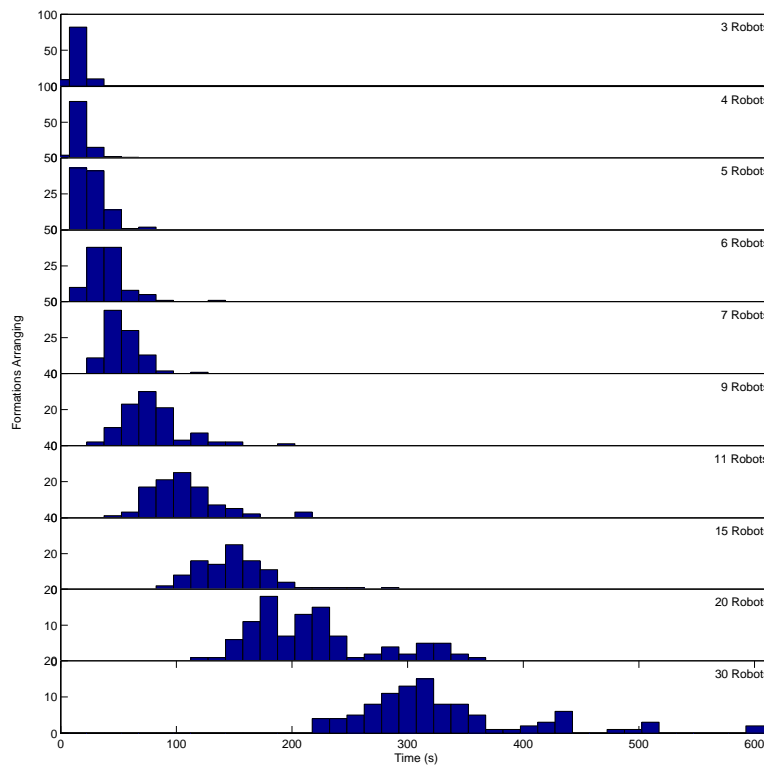


Figure 5.52: Circle formation, Formation arrangement time histograms for 101 simulations for 101 simulations

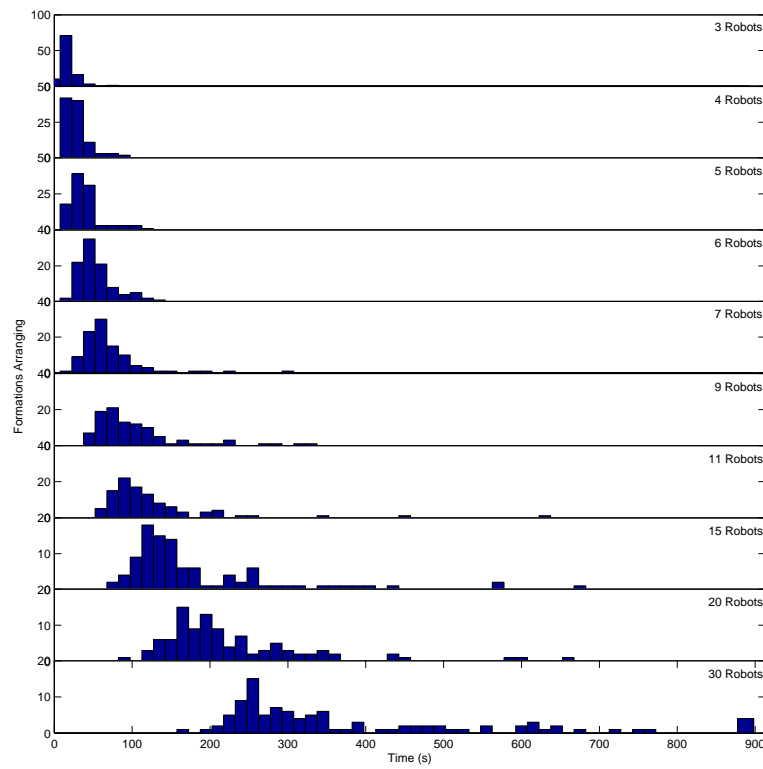


Figure 5.53: Wedge formation, Formation arrangement time histograms for 101 simulations

Formation Arrangement Time Statistics Again, the figures shown in 5.50 to 5.53 do share a common shape with a similar the spreading of the arrangement time, showing a relation with the number of robots in the formation in all the studied formation shapes.

The histograms obtained for the arrangement time shows a similitude to a Bell function, however they cannot fit into ideal bell because the left slope of the bell is not infinite –there is a minimum time value at which a formation can be considered arranged– while the right slope can be infinite –there is not a maximum value for the arrangement time–.

However, the mean arrangement time and the standard deviation of the arrangement times for the different simulations can be studied.

The arrangement times observed in figures 5.50 to 5.53 show a mean and standard deviation values represented in table 5.15. When the resulting values are plot against the number of robots taking part in each formation we obtain the curves shown in figure 5.54.

At figure 5.54 the mean arrangement time shows a linear relationship with the amount of robots building up the formation. Also the standard deviation –the arrangement time spreading– grows with the amount of robots within the formation. The existence of a relationship between the amount of robots in a formation and the time it takes to the robots to build the formation is something to be expected.

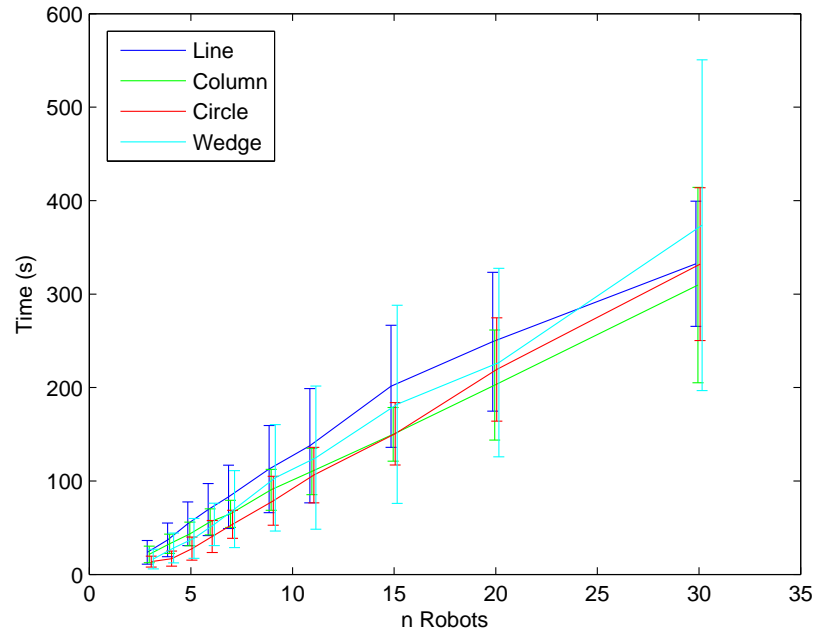
This relationship being linear, and in the same region for all formation shapes, indicate some level of independence between the formation shape and the results achieved by the proposed method.

The mean arrangement time is affected by to main contributions, the time a robot needs to travel from its initial position to a place near the formation and the time spent manoeuvring around the formation. The travel time for a robot between its initial position to some place near the formation is dependent on the initial distance of the robot to the formation. This initial distance is represented by the first parameter in figure 5.48 and can be considered linear to the number of robots.

The obtained result of the mean formation arrangement time being also approximately linear with the number of robots, can indicate that the time spent manoeuvring around the formation is also linear with the number of robots.

The second contribution to the mean arrangement time, the manoeuvring of the robot around the formation, will be ideally zero, when the robot travels directly between the initial point to is insertion point in the formation. However this only can be achieved through a prior planning of the path. The proposed method is reactive, so, considering how the situations have been built, the robot navigates the surroundings of the formation until it founds an empty place to get in, which holds the hypothesis of the manoeuvring time being related with the formation size.

This manoeuvring time probably represents also the main cause of the variation of the standard deviation of the arrangement time with the number of robots involved in the formation. The initial travel time will be mainly influenced by the initial dispersion, which is linear with the amount of robots, and along the initial travel towards the formation little interactions between the robots can be expected, therefore the dispersion on the initial travel time can

Figure 5.54: Formation Arrangement Times, Mean \pm Std

be expected to be low. On the other hand, when the robot reaches the surroundings of the formation, the number of interactions with other robots –the time expended manoeuvring around– will grow with the number of robots and as the number of robots grows higher the interactions are potentially more complex. This increase in complexity can probably explain the higher dispersion on the right slope of the histogram and the resulting deviation obtained from the data.

Table 5.15: Formation Arrangement Times for the different Formation Shapes and number of robots used in the simulation, without noise

	Line	Column	Circle	Wedge
3	24 \pm 12	21 \pm 8	14 \pm 5	16 \pm 10
4	37 \pm 18	33 \pm 10	17 \pm 8	28 \pm 16
5	54 \pm 23	43 \pm 12	28 \pm 12	39 \pm 21
6	69 \pm 27	56 \pm 13	41 \pm 17	54 \pm 22
7	83 \pm 33	65 \pm 14	54 \pm 15	70 \pm 41
9	113 \pm 46	91 \pm 21	79 \pm 26	103 \pm 56
11	138 \pm 61	110 \pm 24	106 \pm 29	125 \pm 76
15	201 \pm 65	150 \pm 28	150 \pm 33	182 \pm 105
20	249 \pm 74	203 \pm 58	219 \pm 55	229 \pm 98
30	332 \pm 66	310 \pm 104	332 \pm 81	374 \pm 176

Noisy scenarios The same set of tests done with ideal conditions has been repeated in scenarios where the position and orientation of the robots is subjected to Gaussian noise –the one expected when using real sensors and actuators–. The used noise is normally distributed, with zero mean and a variance of $r_{Lim}/2$ for the position measurements and a variance of $1/30\pi(rad)$ for the orientation measurements. The noise in the actuators is applied as a uniform distribution of $\pm 5\%$ around the given reference speed. This noise levels used in these simulations are above the maximum noise observed in the real robot tests. The same number of simulations as in the noiseless case has been done to obtain the noisy results.

Distance To Formation Figures 5.55 to 5.58 show the results of the mean distance to formation and its standard deviation for the noisy scenarios, these results can be compared to the noiseless ones shown in figures 5.44 to 5.47.

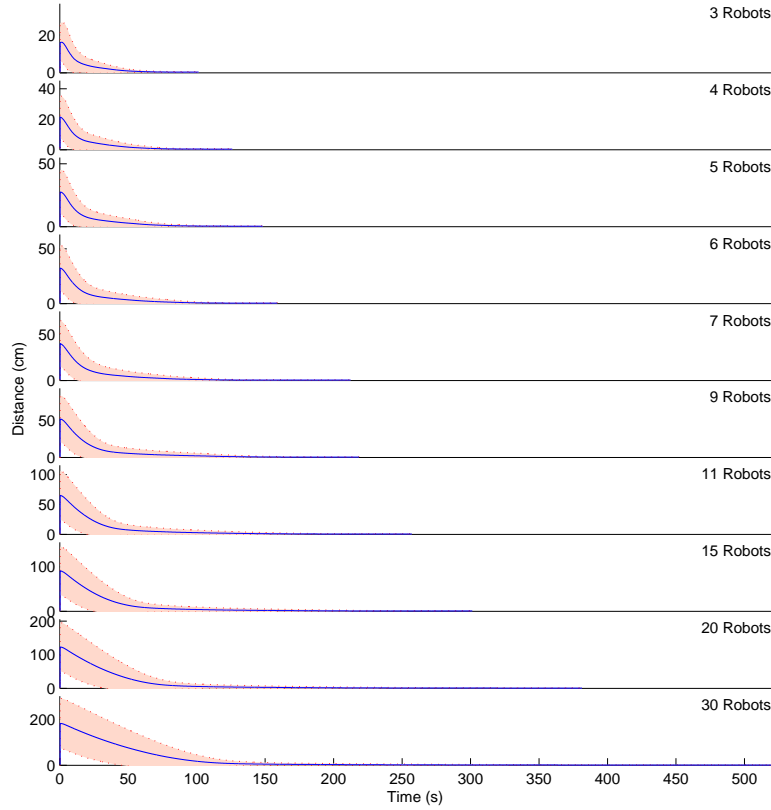


Figure 5.55: Line formation, Mean distance to formation along simulation time \pm Std, Noisy simulation

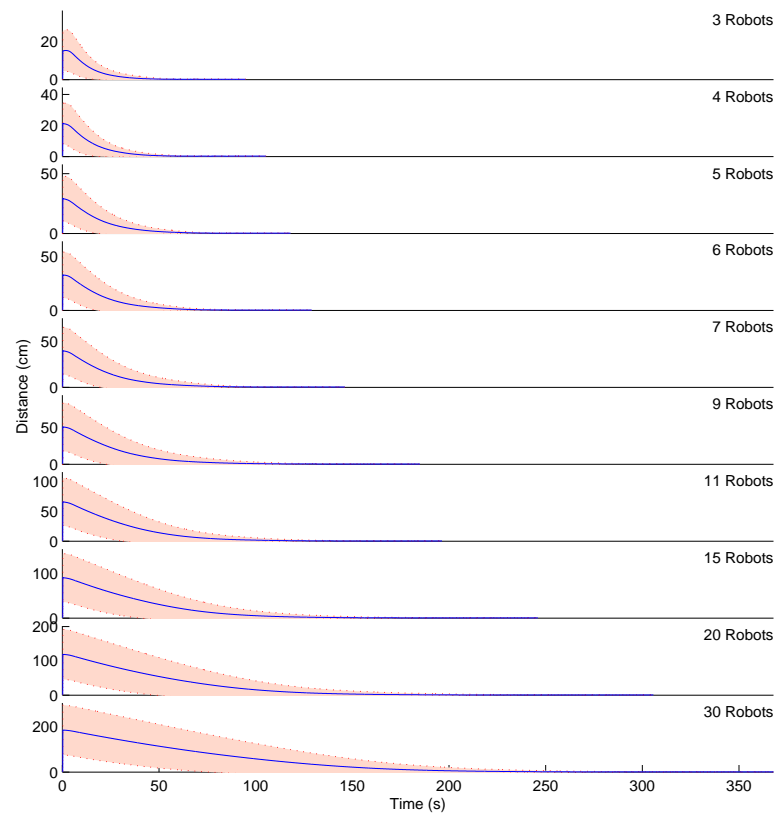


Figure 5.56: Column formation, Mean distance to formation along simulation time \pm Std, Noisy simulation

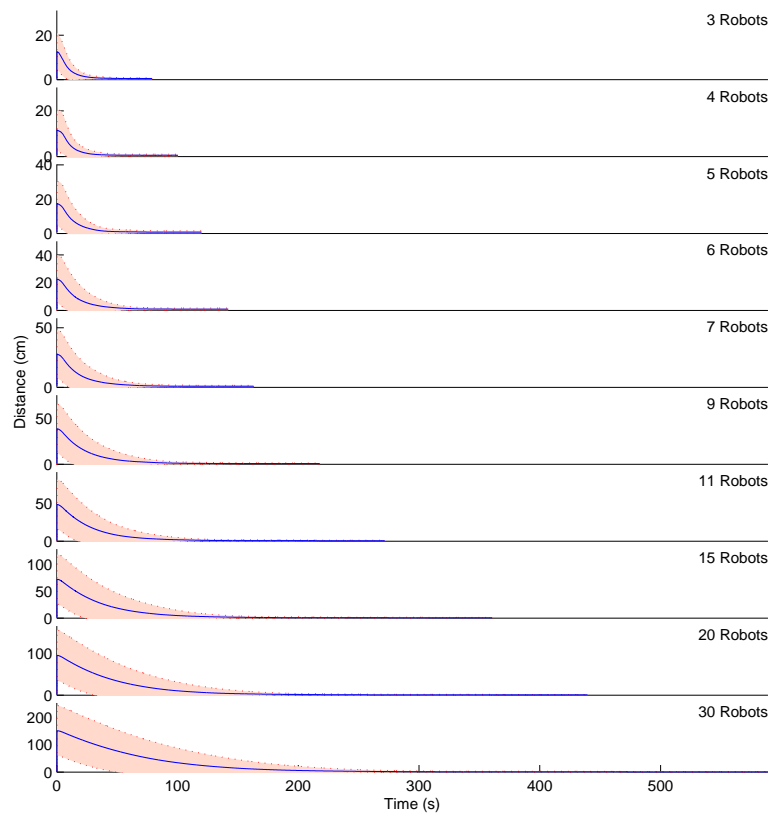


Figure 5.57: Circle formation, Mean distance to formation along simulation time \pm Std, Noisy simulation

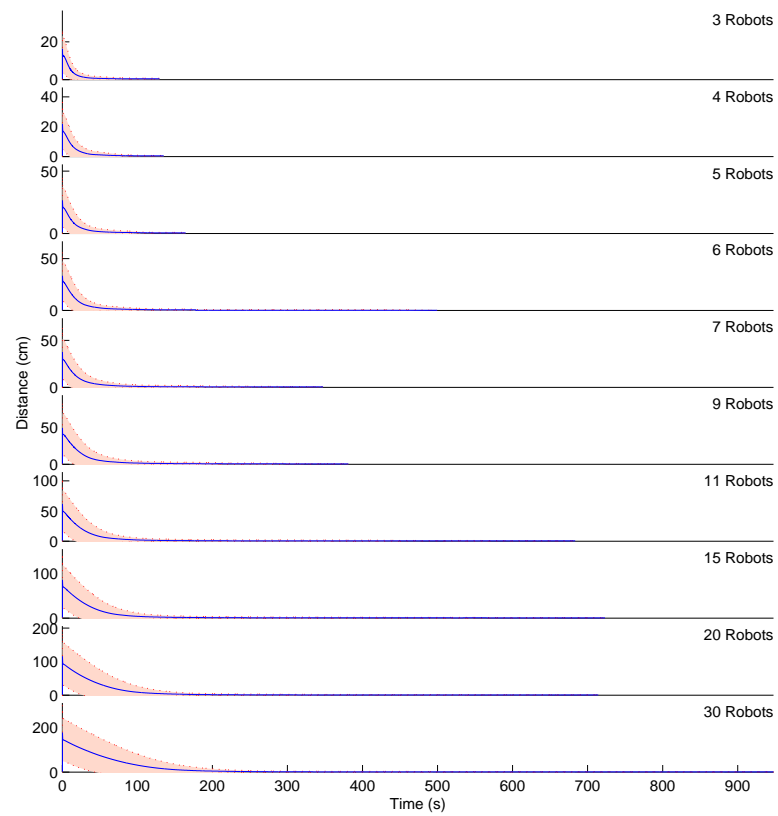


Figure 5.58: Wedge formation, Mean distance to formation along simulation time \pm Std, Noisy simulation

Distance To Formation Fit The results for the noisy scenarios related with the distance between robots and formation do look like the results obtained for the noiseless scenarios. The fitting of the noisy scenarios with the exponential decay function are shown in figures 5.59 and 5.60. Also, figures 5.61 and 5.62, show the the results form the noiseless and the noisy simulations together. It can be seen that all the fitting parameters are quite similar for both test sets, even it shows slightly better results in some cases at the noiseless simulations, where the noise helps to solve symmetry equilibrium situations quicker.

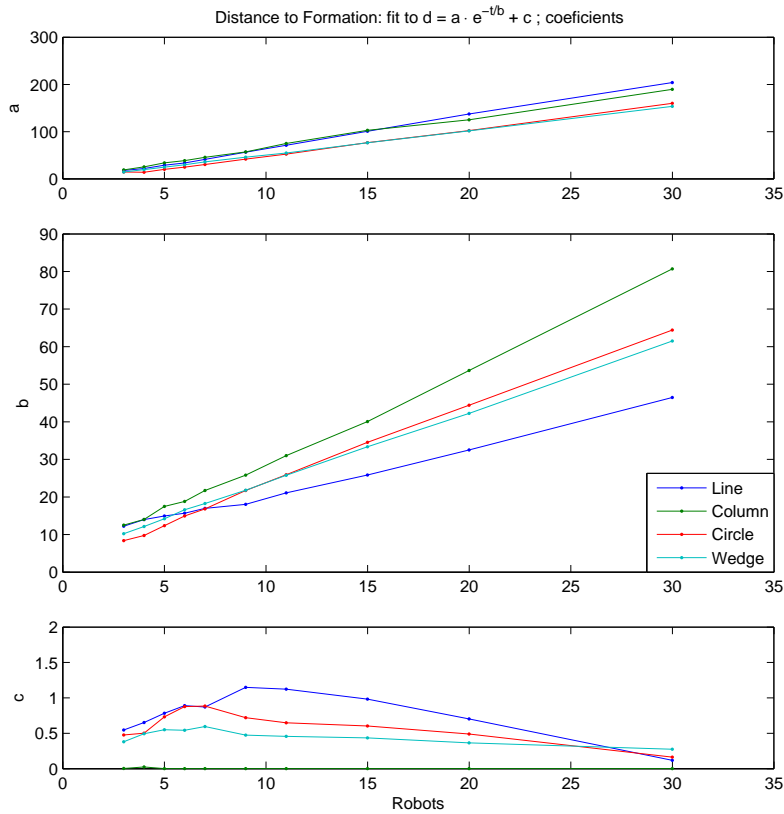


Figure 5.59: Coefficients from the fitting of the distance to formation along simulation time to an exponential function, Noisy simulation

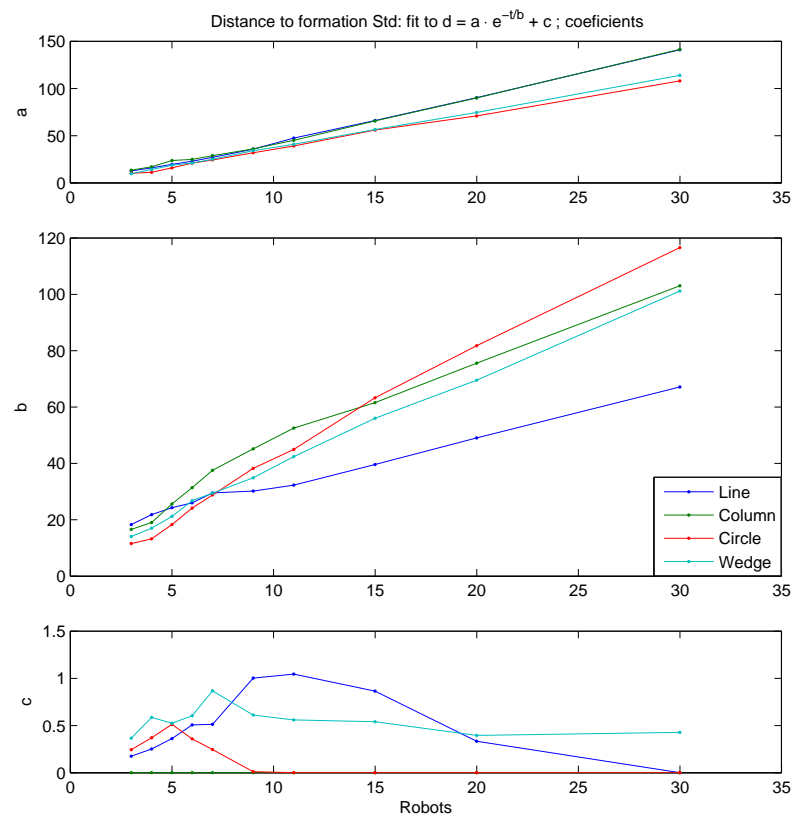


Figure 5.60: Coefficients from the fitting of the standard deviation on the distance to formation along simulation time to a exponential function, Noisy simulation

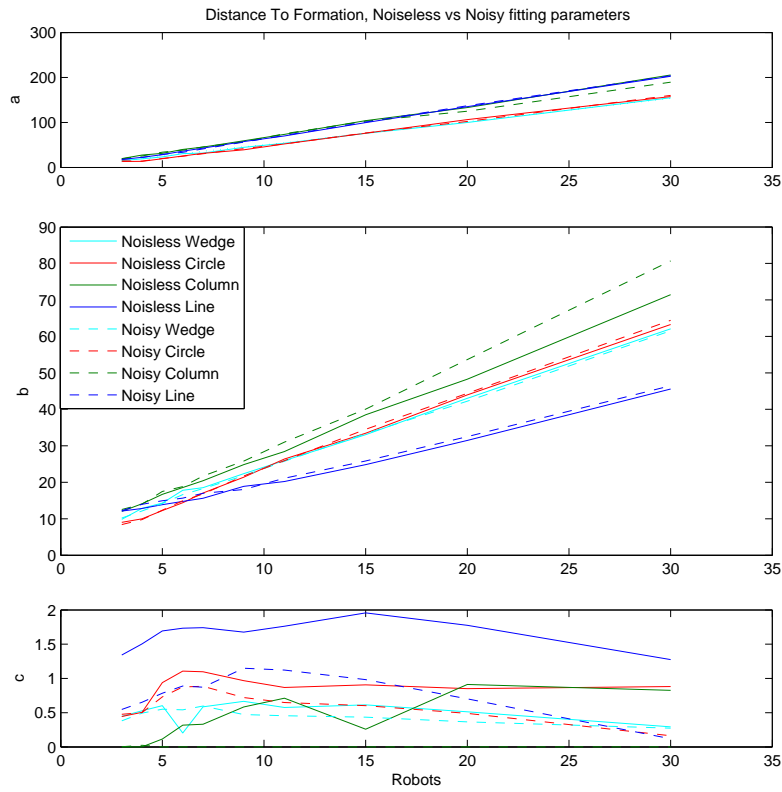


Figure 5.61: Coefficients from the fitting of the distance to formation along simulation time to an exponential function, Noisy Vs Noiseless, simulation

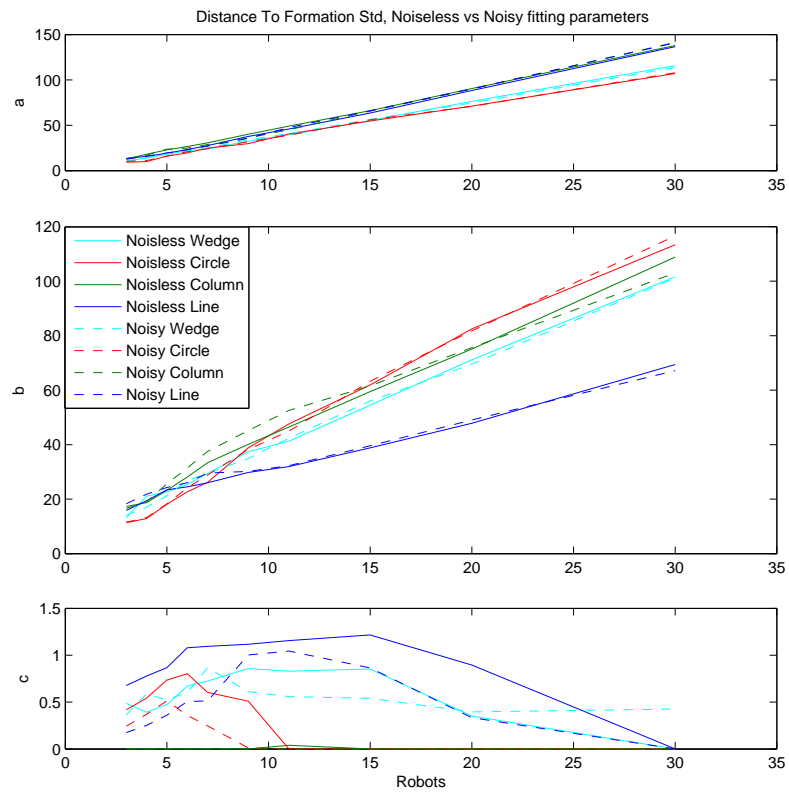


Figure 5.62: Coefficients from the fitting of the standard deviation on the distance to formation along simulation time to a exponential function, Noisy Vs Noiseless, simulation

Formation Arrangement Time The arrangement time histograms for noisy scenarios are shown in figures 5.63 to 5.66 which are quite similar to their noiseless counterparts shown in figures 5.50 to 5.53.

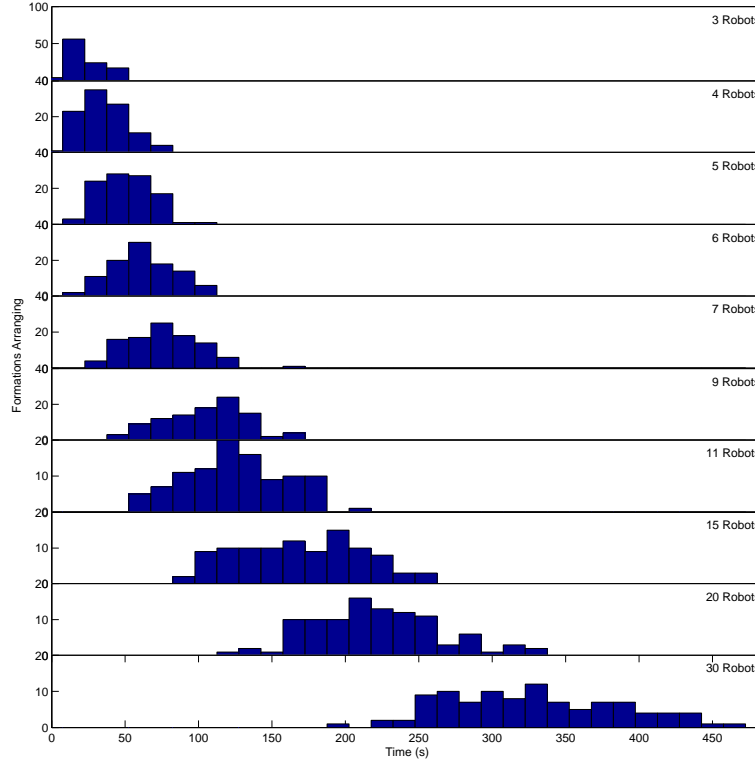


Figure 5.63: Line formation, Formation arrangement time histograms, Noisy simulation

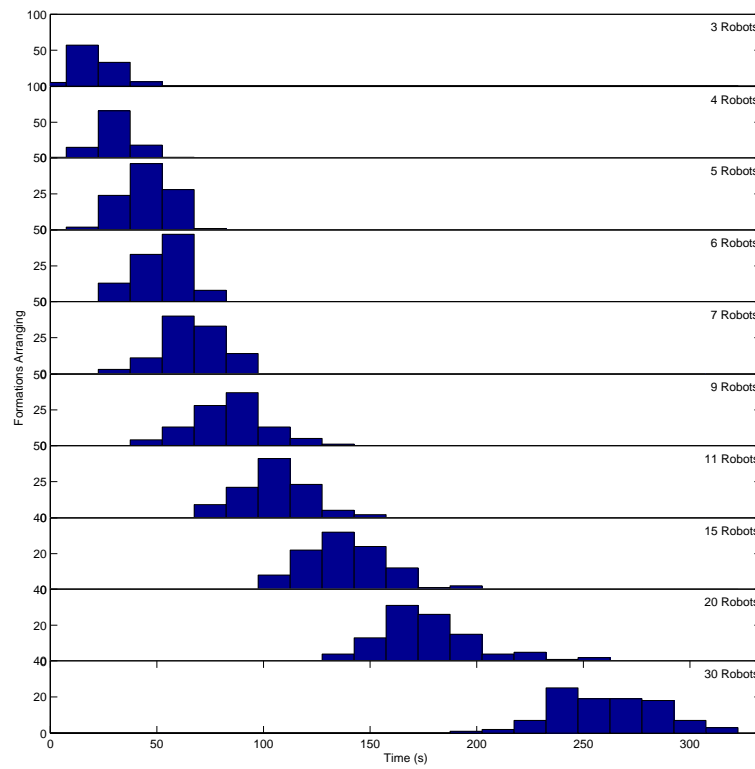


Figure 5.64: Column formation, Formation arrangement time histograms, Noisy simulation

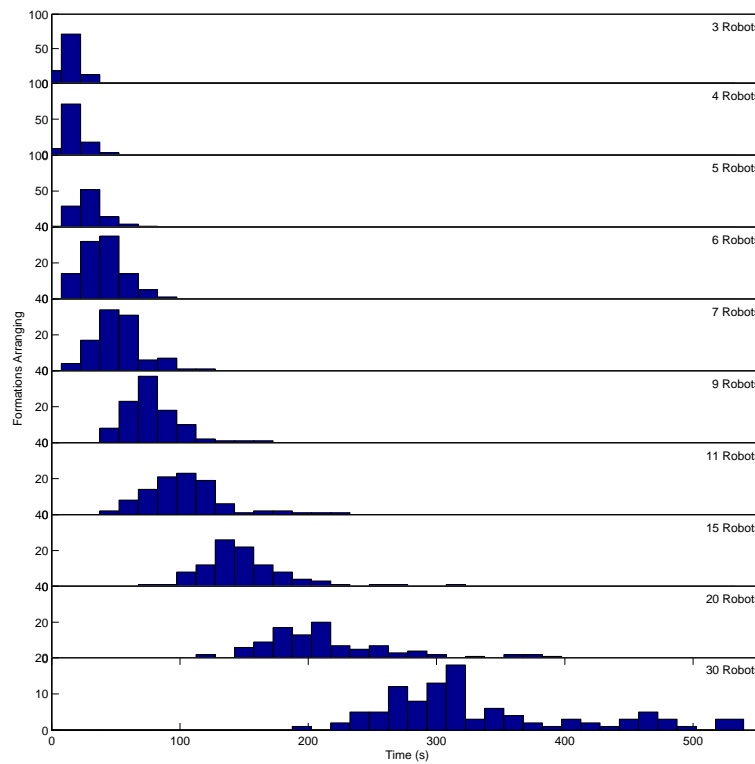


Figure 5.65: Circle formation, Formation arrangement time histograms, Noisy simulation

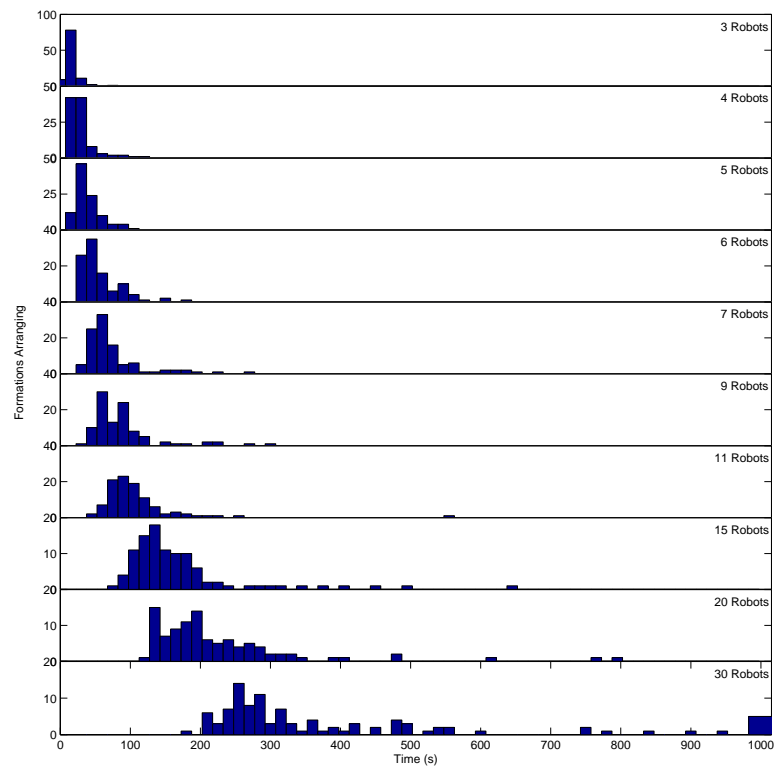


Figure 5.66: Wedge formation, Formation arrangement time histograms, Noisy simulation

Formation Arrangement Time Statistics Again, using the obtained histograms, the mean and standard deviation of the arrangement time for the studied formations are shown in table 5.16 and depicted in figure 5.67. The observed results for the noisy scenarios are quite similar to those obtained in the noiseless scenarios.

Table 5.16: Formation Arrangement Times for the different Formation Shapes and number of robots used in the simulation, Noisy simulation

	Line	Column	Circle	Wedge
3	23 ± 12	21 ± 9	13 ± 6	17 ± 9
4	36 ± 16	31 ± 9	17 ± 9	29 ± 19
5	50 ± 18	45 ± 11	29 ± 12	40 ± 19
6	62 ± 21	53 ± 11	41 ± 15	57 ± 29
7	76 ± 24	66 ± 13	51 ± 18	74 ± 42
9	105 ± 28	84 ± 17	78 ± 21	88 ± 46
11	125 ± 33	105 ± 15	104 ± 32	108 ± 58
15	169 ± 42	138 ± 18	150 ± 35	168 ± 87
20	220 ± 43	177 ± 23	215 ± 53	227 ± 113
30	328 ± 58	261 ± 24	333 ± 76	398 ± 275

Comparing the results of the noiseless scenarios with the noisy ones, the resulting parameters are within the same range, so we can state that the proposed method and specific situations/actions are quite resistant to noise disturbances.

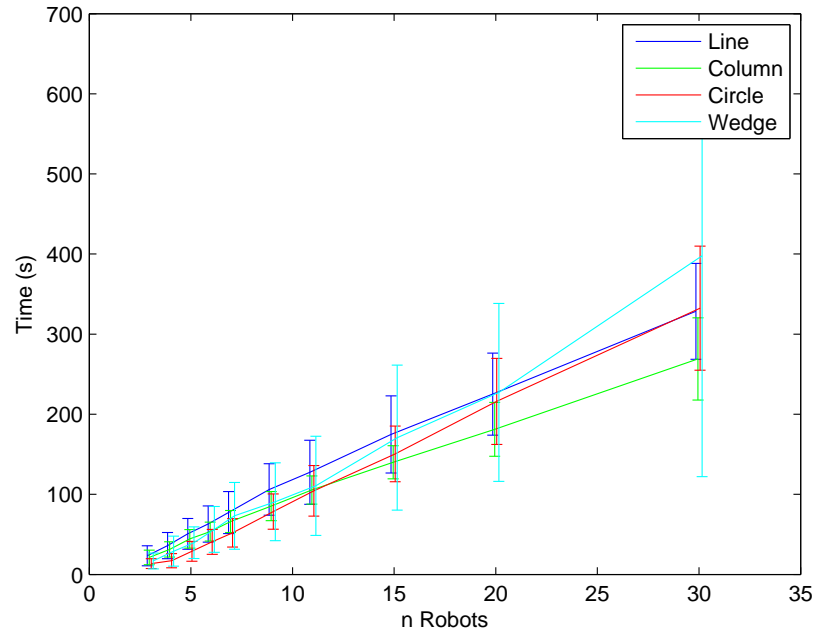


Figure 5.67: Formation Arrangement Times, Mean \pm Std, Noisy simulation

Formation Manoeuvring

The reactive nature of the method allows, without any change or previous warning to the robots, to modify the leading direction on the fly, changing so the formation moving direction. This changes in the formations movement direction can be considered as manoeuvring the formation as a single entity. However it should be considered that when the changes in direction of the formation are sharp enough the formation will be disrupted because the robots are not able to turn or move fast enough to keep their place inside the formation.

In order to test the ability of the robots to act as a single entity, an already stabilised formation is going to be subjected to a $\pi/2$ turn at different turn rates. This allows us to observe which formation shapes are more resistant to manoeuvring and how the proposed collection of situation-actions handles the stress of the manoeuvres.

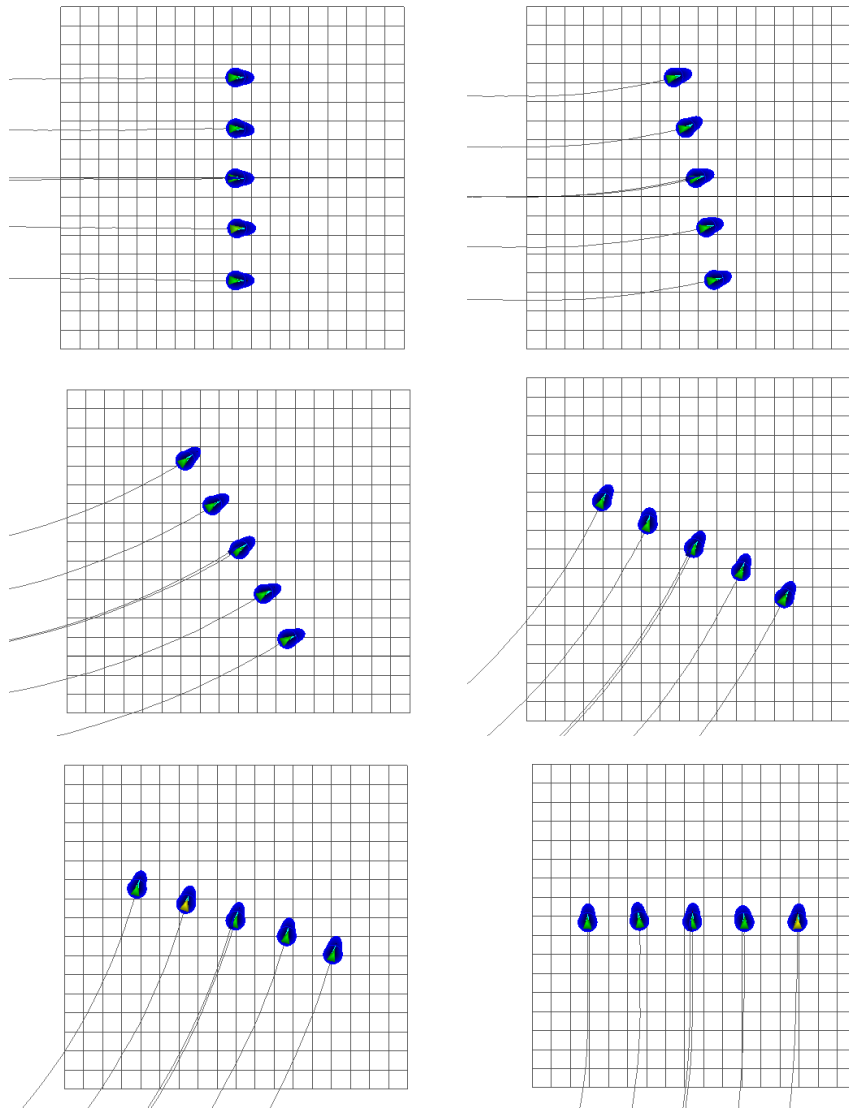
To illustrate the robot navigation capabilities and the formation manoeuvring, some snapshots taken along the simulations are shown.

Along the captures shown in figure 5.68 a slow turn of a column formation can be observed, at a turn rate of 0.01rad/s in the leading direction of the formation field. In this test the robots are able to follow the formation shape without nearly any disruption, ending the turning manoeuvre without need to rearrange the formation.

In figure 5.69 a medium speed turn of the same formation can be observed, now with a turn rate of 0.05rad/s . The robots are barely able to follow the formation field without disrupting its shape, the two robots on the tips of the formation are not able to follow it. Once the turning manoeuvre ends, with the two robots out of shape, the arranging process begins and the two scattered robots find new places inside the formation.

A very sharp turn can be seen along the captures in figure 5.70. With a turn rate of 1rad/s —above the maximum turn rate of the individual robots— the manoeuvre is so quick that the robots do not even begin to turn as a formation, but they can only turn individually at their current positions and immediately after, a full arrangement process starts, which needs to be complete in order to let the formation take shape again.

The previous behaviours can be observed in any formation shape. Depending on the formation wide and length, the formations begin to disrupt at different turning rates of the leading direction, but after the turn manoeuvre ends, the rearrangement process takes place again and the formation is re-established. One exception, not completely unexpected, is the circle formation. As can be seen in the captures on figure 5.71, were 11 robots perform a turn at 1rad/s on the leading direction, the formation shape is kept along the manoeuvre with only minor disruptions. The circle formation is kept without disruption during the turning manoeuvres while the manoeuvre turn rate is below the individual robot turn rate. When the turn rate of the manoeuvre is above the individual turn rate, each robot will turn at its maximum rate and so will do the formation—the manoeuvring turn rate is saturated but the formation does not lose shape—

Figure 5.68: Column formation, slow turn (0.01rad/s) for 5 robots

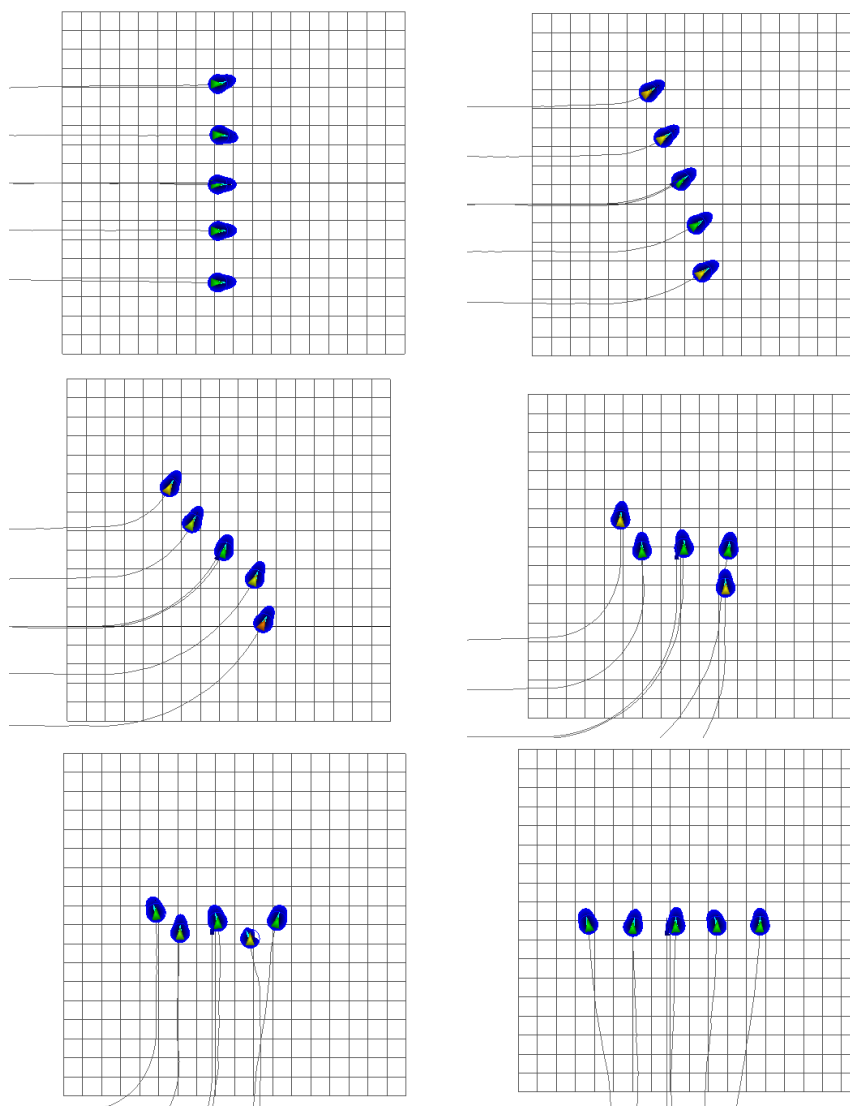
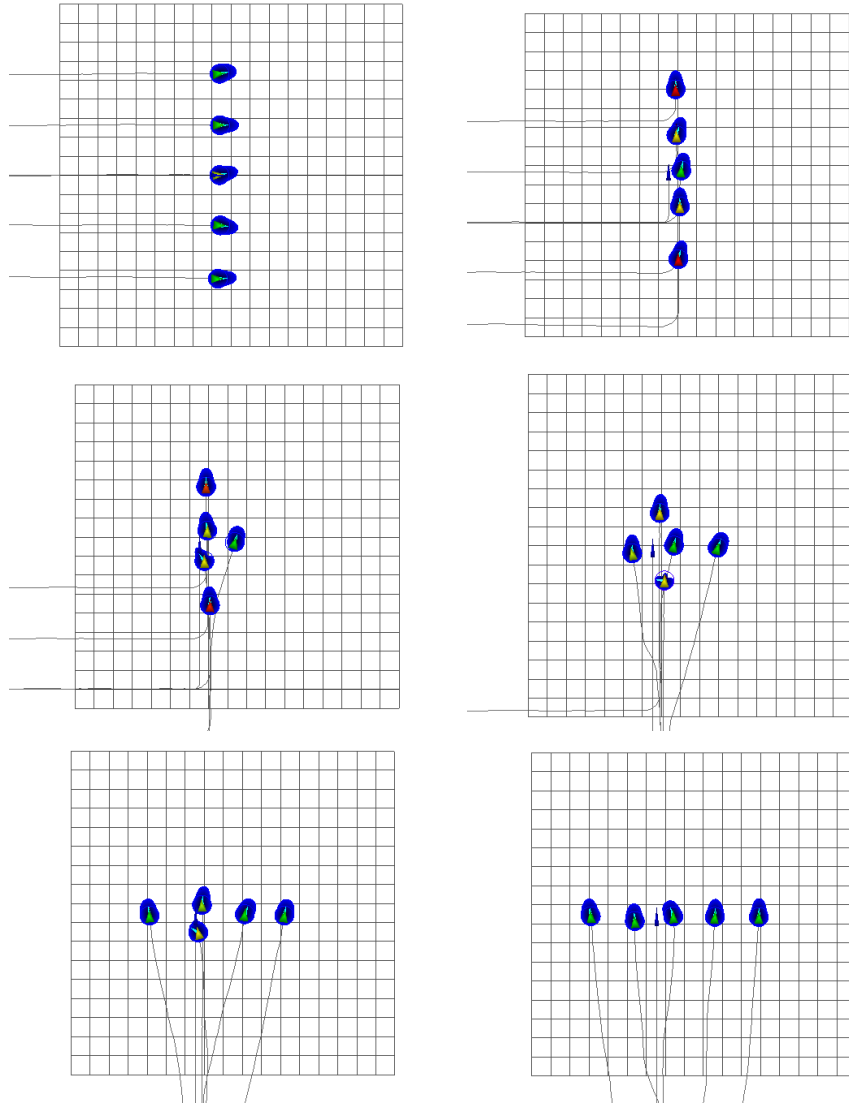


Figure 5.69: Column formation, medium speed turn (0.05 rad/s) for 5 robots

Figure 5.70: Column formation, sharp turn (1rad/s) for 5 robots

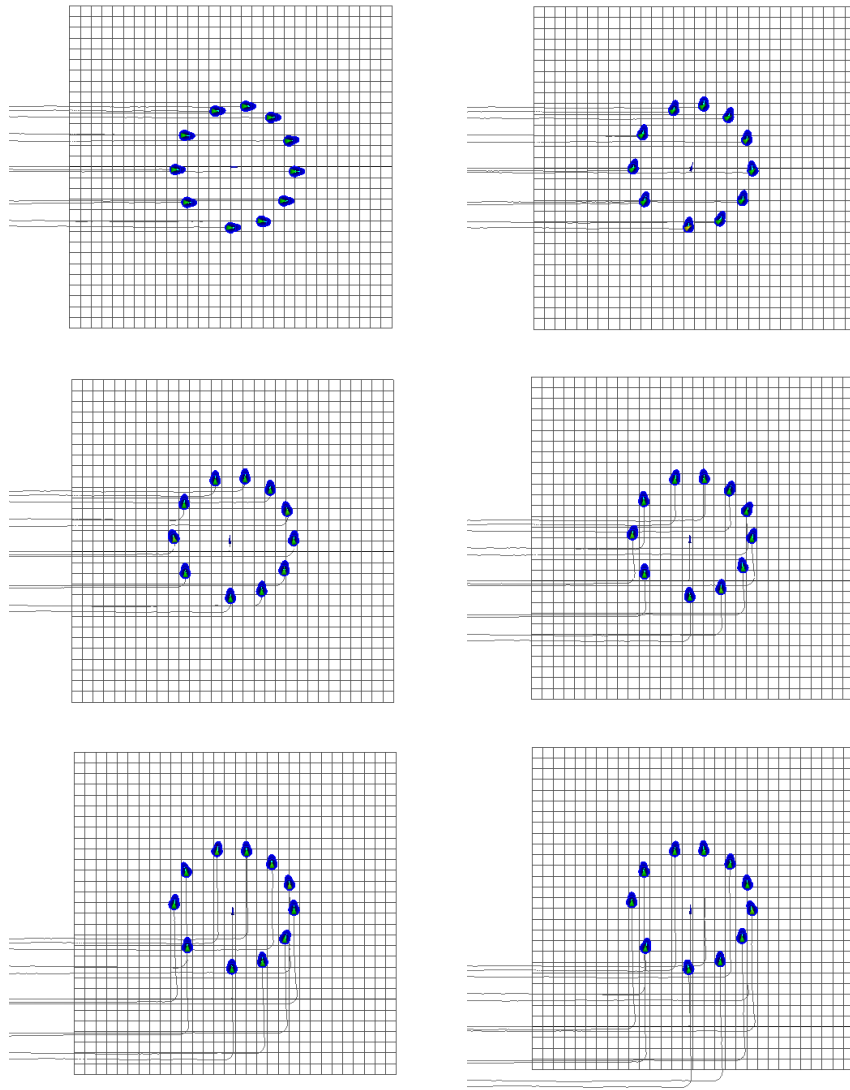


Figure 5.71: Circle formation, sharp turn (1rad/s) for 11 robots

The arrangement tests applied to the formation manoeuvring simulations, indicate that the robots will rearrange the formation following an exponential decay of the distance to the formation along time. Again a statistically significant amount of tests, 100 random initial positions for each formation shape and turn ratio, have been done to study the general behaviour of the turning manoeuvres. Only the results for five robots formations are shown, other formation sizes follow the same general behaviour.

When observing the distance to formation along the turning manoeuvres shown in figures 5.72 to 5.75 three distinctive regions can be pointed. The first 30 seconds corresponds to the initial stable formation moving in the initial direction, 0 rad. After that, it comes the turning manoeuvre itself, which duration depends on the turning rate of each test $\approx 157s$ for 0.01 rad/s to $\approx 1,57s$ for 1 rad/s. Finally, after the turn manoeuvre ends, the arrangement process works to achieve again an stable configuration.

On the first region, along the first 30 seconds, a low distance, low dispersion was to be expected since it is the evolution of an already established formation.

On the second region, where the turning manoeuvre takes place, two different shapes of the standard deviation of the distance to formation can be observed, which can be associated with two dispersion processes.

The first process, which can be seen in the slow turn, 0.01rad/s, at all the formation shapes. The 'dispersion' is kept mostly constant along the manoeuvre time. This is an indicator that the robots are not perfectly inside the formation but they are able to follow the manoeuvre without losing the formation shape.

In the second dispersion process, the 'dispersion' grows along the manoeuvring period, indicating that the robots are not able to keep the formation shape along the manoeuvre.

When handling the formation as a single entity, the evolution of the dispersion along the turning manoeuvre can be used as indicator to adjust the turning rate, and so keep the robots in shape along the manoeuvre.

On the last region, the arrangement period, it can be observed that the initial assumption that any disruption of the formation will be handled as an arrangement process, is satisfied. The mean distance to formation and the distance standard deviation follow the already familiar exponential decay.

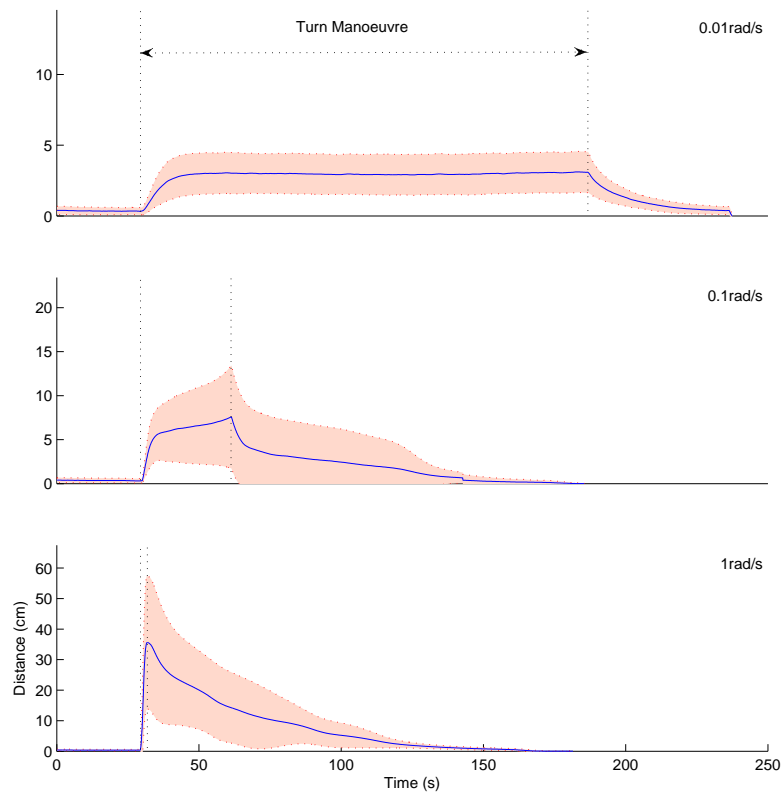


Figure 5.72: Distance to formation for various turn ratios, 5 robots in line formation.

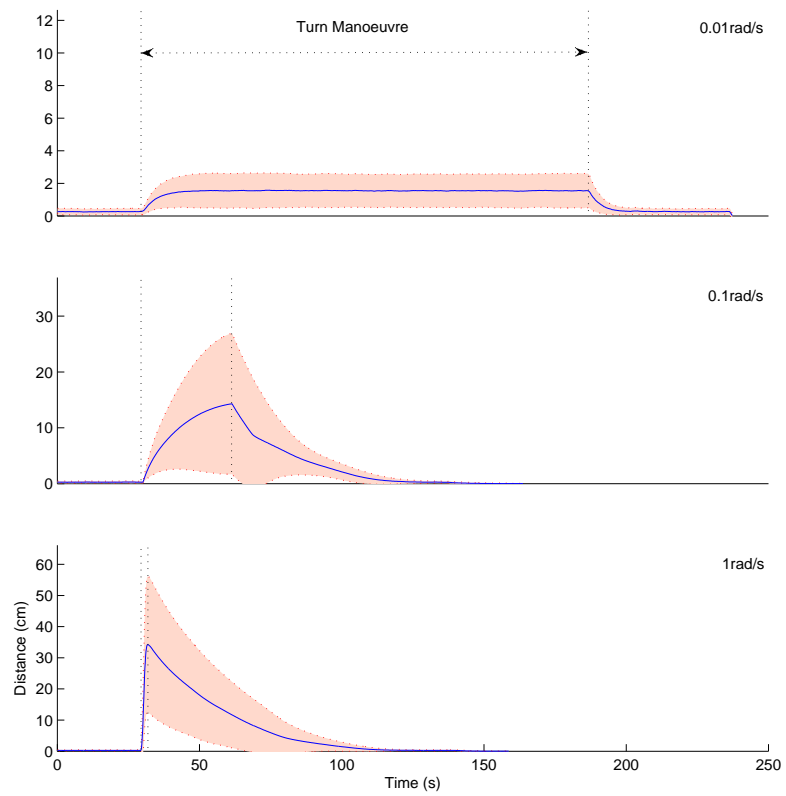


Figure 5.73: Distance to formation for various turn ratios, 5 robots in column formation.

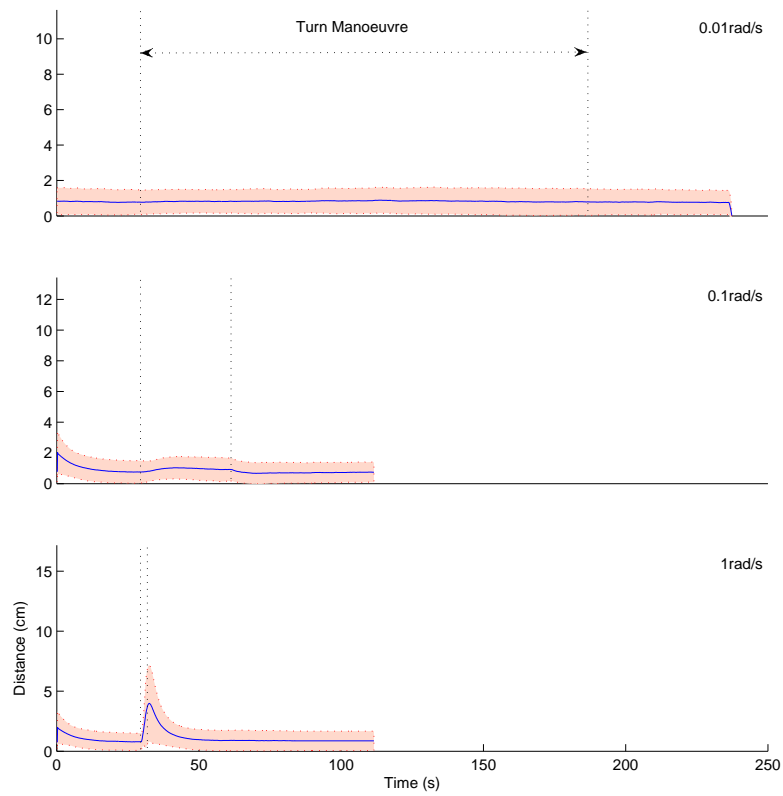


Figure 5.74: Distance to formation for various turn ratios, 5 robots in circle formation.

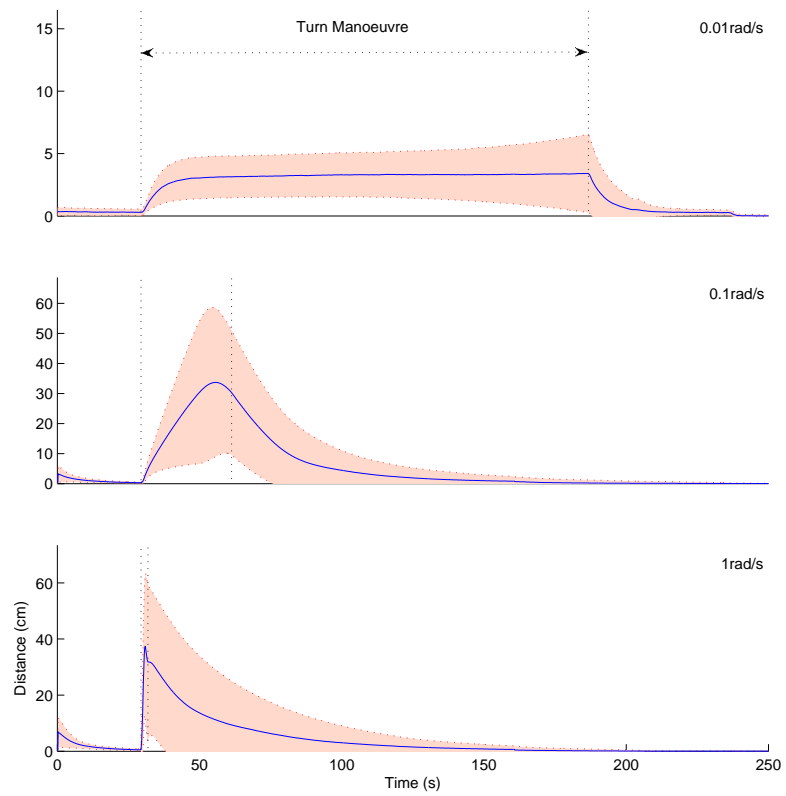


Figure 5.75: Distance to formation for various turn ratios, 5 robots in wedge formation.

When the arrangement period is analysed to fit the exponential decay the observed parameters are not the same as the one obtained for the initial random arrangement. The first fitting parameter, the initial distance to the formation is not comparable with the one obtained for the random arrangement because the initial state of both experiments is not the same.

The second fitting parameter, the exponential time constant should show similar results as in the case of the initial random arrangement process. The obtained values for the time constant after the turn manoeuvre fall in the same region of the results obtained for the random initialization process, however the arrangement process at the manoeuvring tests are quicker than the one observed at the random initialization process.

An explanation for this results is that along the initial arrangement tests the robots were randomly distributed over a large enough area, on the case of the turning manoeuvre, the distribution of the robots at the beginning of the arrangement process is far from randomly distributed, on the contrary, they are in a very similar organized structure at each test.

In these results the effects of the approximation process can be considered negligible, since the initial position of the robots after the manoeuvre is already a compact structure near the formation center. The main contribution to the time constant is then associated to the manoeuvring period of the arrangement process, discussed at the formation initialization analysis.

Formation Switching

In general, a reactive navigation method allows to change the conditions imposed on the formation, these changes are handled in a transparent way without any especial actuation.

This ability to handle changes in the formation can be illustrated with the change of the formation from an already established shape into another. The formation shape does not need to be constant along time, it can be switched on the fly without the need to add or modify anything to the already shown set of situations and actions; the robots need only to be informed of the new shape to use it in their evaluation.

Some traces of formation switching processes are shown in figures 5.76 to 5.79. While in these examples only transitions between full formations are shown, they can be used also to get an idea about how modifications on the formation shape –not complete switches– can be employed to better guide the robots as a single entity, i.e. bending a line formation on a sharp turn or practice an encircling manoeuvre, can be smoothly achieved only by setting how the shape evolves along time.

The distances to formation traces are not shown here because they are nearly equal to those shown in the sharp turning manoeuvres; the formation field sensed by each robot changes so abruptly that no transition manoeuvring is performed by the robots, but a full rearrangement.

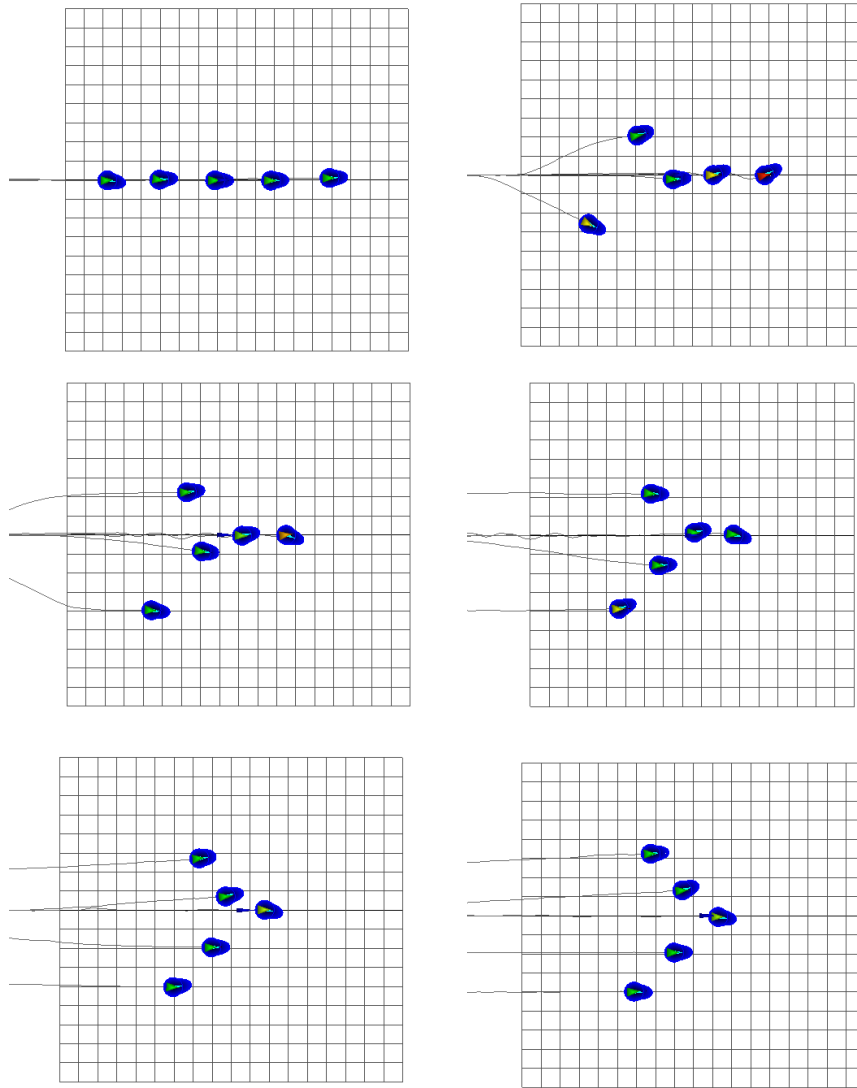


Figure 5.76: Formation Transition, Line to Wedge formation with 5 robots

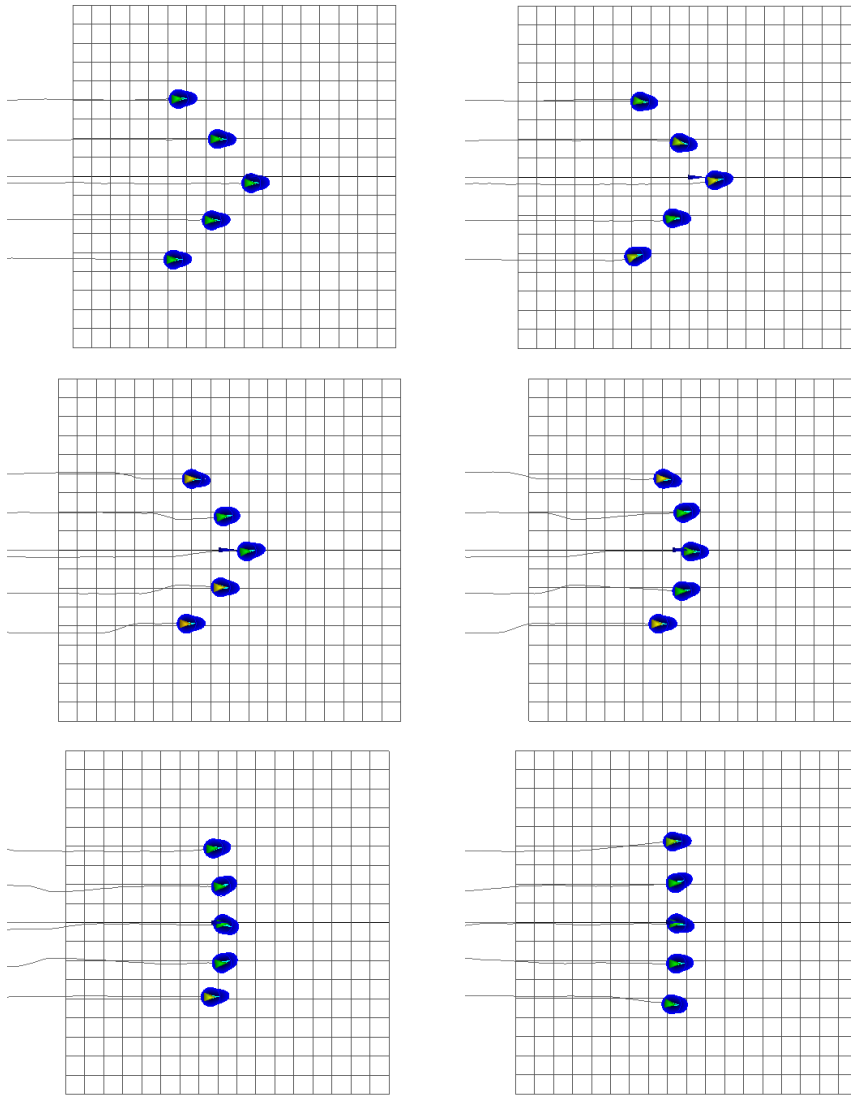


Figure 5.77: Formation Transition, Wedge to Column formation with 5 robots

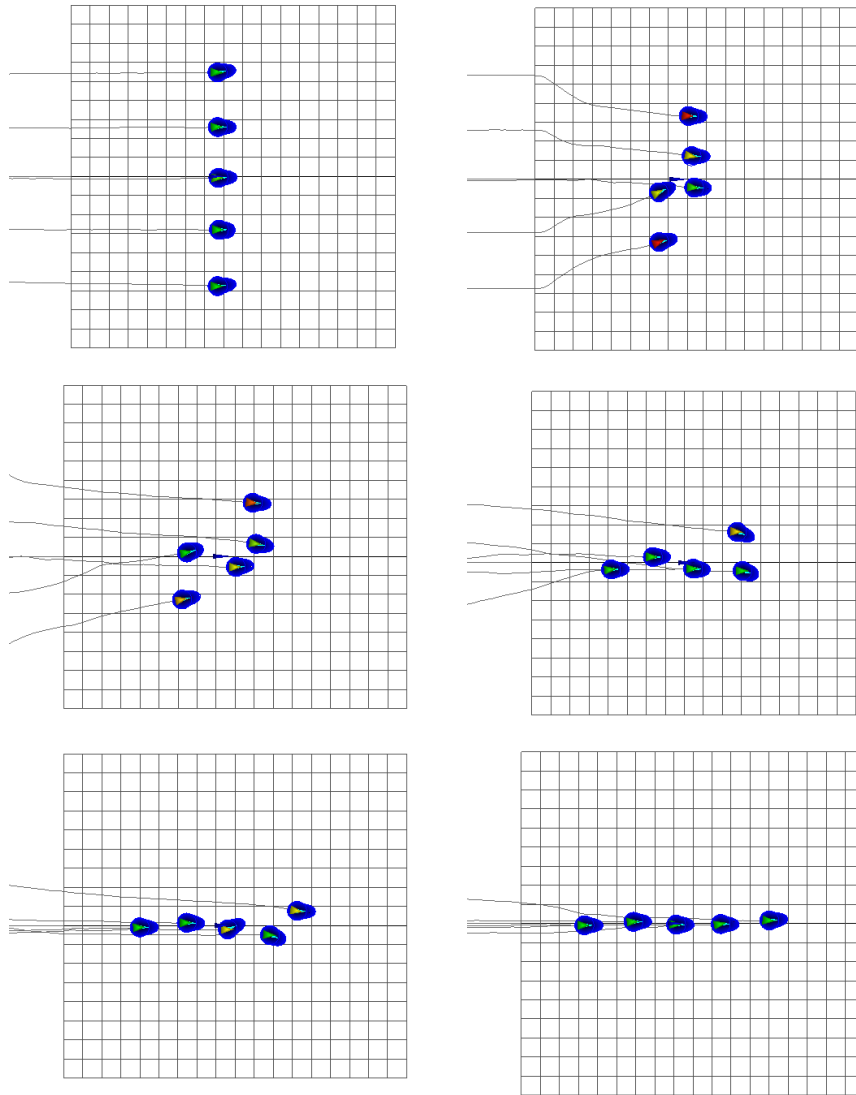


Figure 5.78: Formation Transition, Column to Line formation with 5 robots

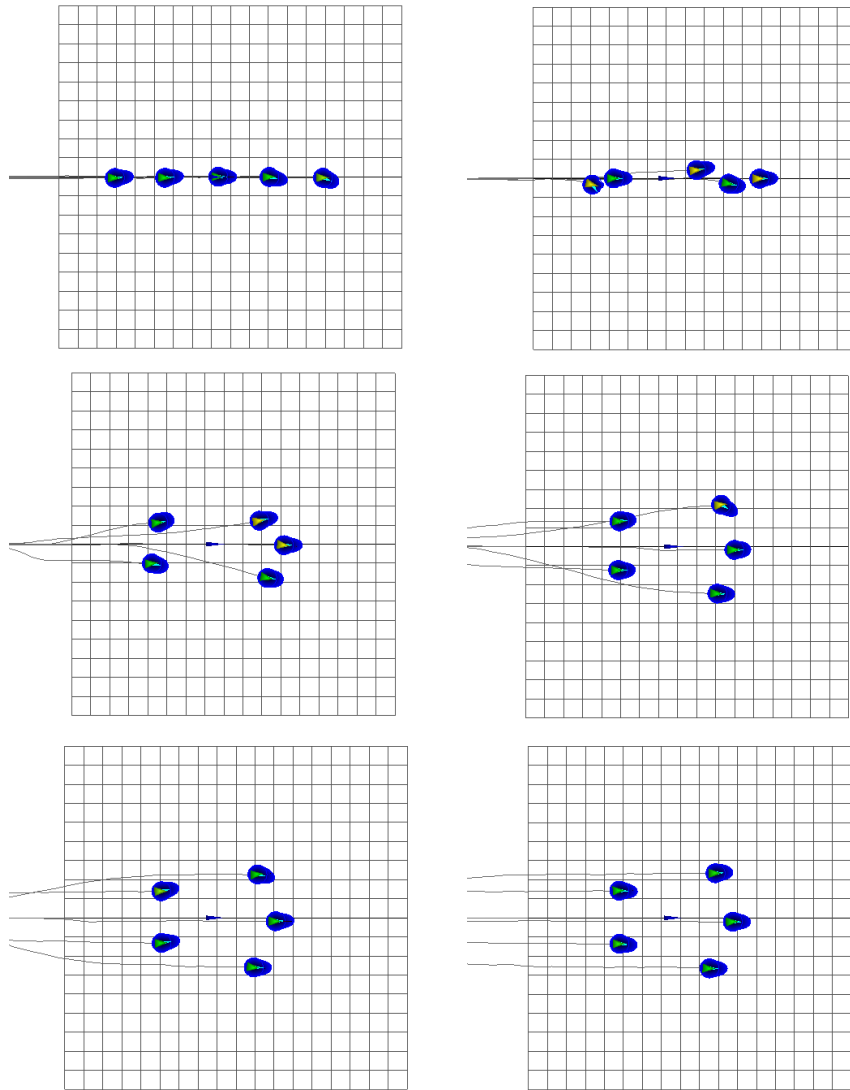


Figure 5.79: Formation Transition, Line to Circle formation with 5 robots

Formation and Obstacles

The obstacle avoidance used by the individual robots in the formation is handled by the situation $S_{1.1}$, and employs the same evaluation and actions used for the unorganized robot group case. This simple obstacle avoidance method is enough to provide each individual robot the capability of avoid collisions with static elements in the scenario. Along the obstacle avoidance, the formation-related actions are still active, however the forbidden action used to avoid the collision with the obstacles discards any other action which leads the robot towards and obstacle. This is done without using any explicit priority mechanism that deactivates the formation-related actions in the presence of an obstacle, the priority mechanism is implicit in the way that the *Forbidden* actions and the *Elect* are combined in the action maps analysis.

That the formation-related actions remain active along the obstacle avoidance process, means that the robot will move again towards the formation as early as the formation direction is clear again and only those robots directly influenced by the obstacle will be affected.

Another interesting aspect is that the formation positioning algorithm can be set to move only along the leading direction. This means that the movement of the formation field coordinates origin will not be affected by the presence of the obstacle. The formation trace will transverse the obstacle, only the robots will skirt the obstacle and, when clear from it, they will return to the original formation as if no obstacle was present. Those effects can be seen in figure 5.80

The presence of obstacles big enough to affect most of the formation at once, as the images shown in figure 5.81, is handled in the same way.

The effects of those big obstacles affect all the robots of the formation through the combination of several situation-actions sets.

Those robots directly in the proximity of the obstacle will perform the avoidance manoeuvre as ever.

On the other hand, the robots not directly in the proximity of the obstacle will be affected by the lack of movement of the other robots through the formation positioning algorithm.

The formation positioning algorithm will adapt to the movement of those robots currently avoiding the obstacle by slowing the formation movement. This will force the robots not directly affected by the obstacle to also slow their movement, and wait for the robots avoiding the obstacle.

Of course, the robot-robot collision avoidance mechanism remains active along all the process so the robots will avoid colliding each others.

The obstacle avoidance only affects the formation structure in the moment when the obstacle is in the path of the robots, if the obstacle does not affect the robot movement, the robots keep the formation structure as can be seen in 5.82.

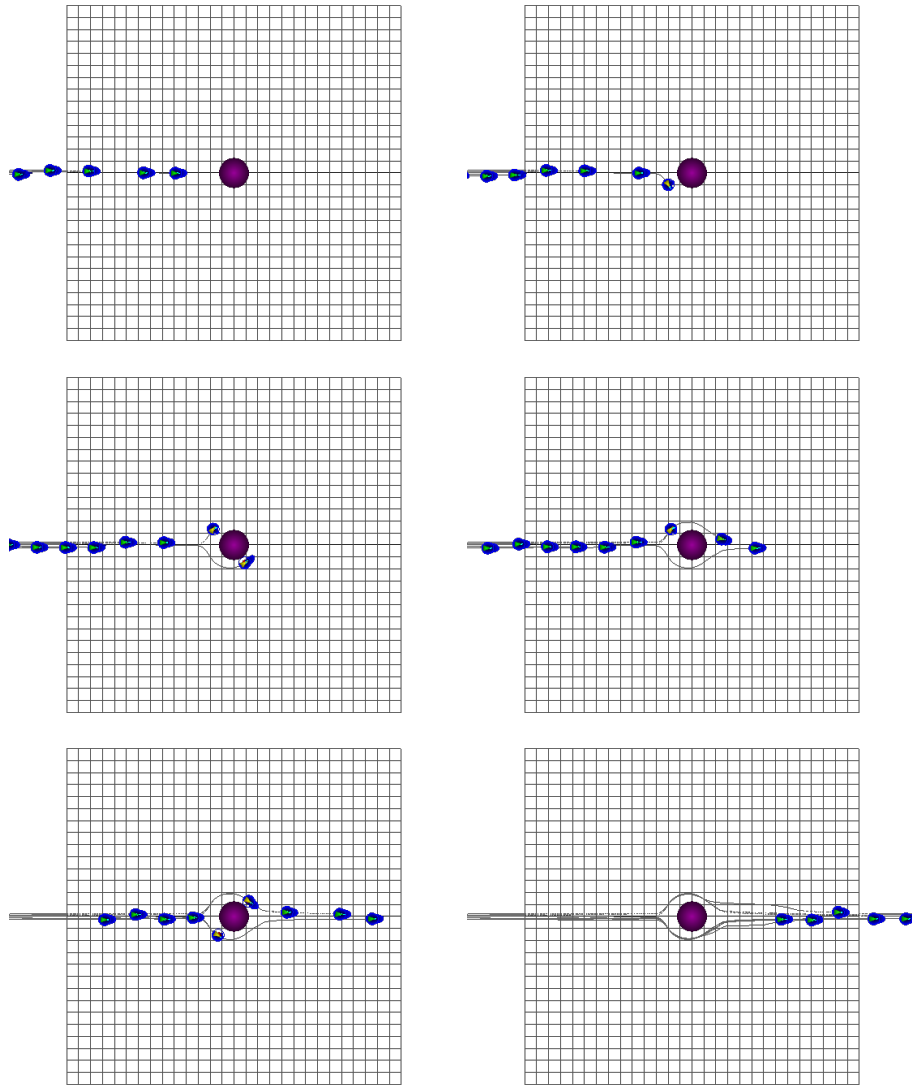


Figure 5.80: Line formation vs. Column obstacle with 9 robots

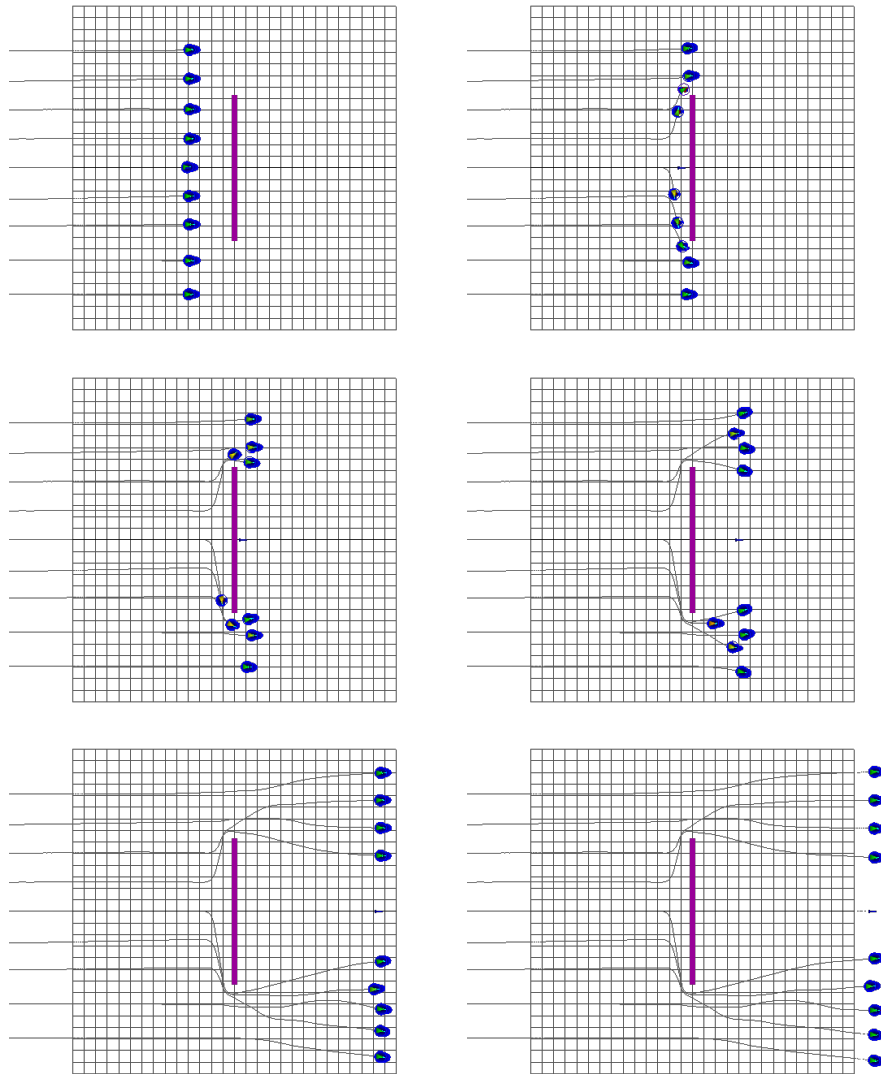


Figure 5.81: Column formation vs. Wall obstacle with 9 robots

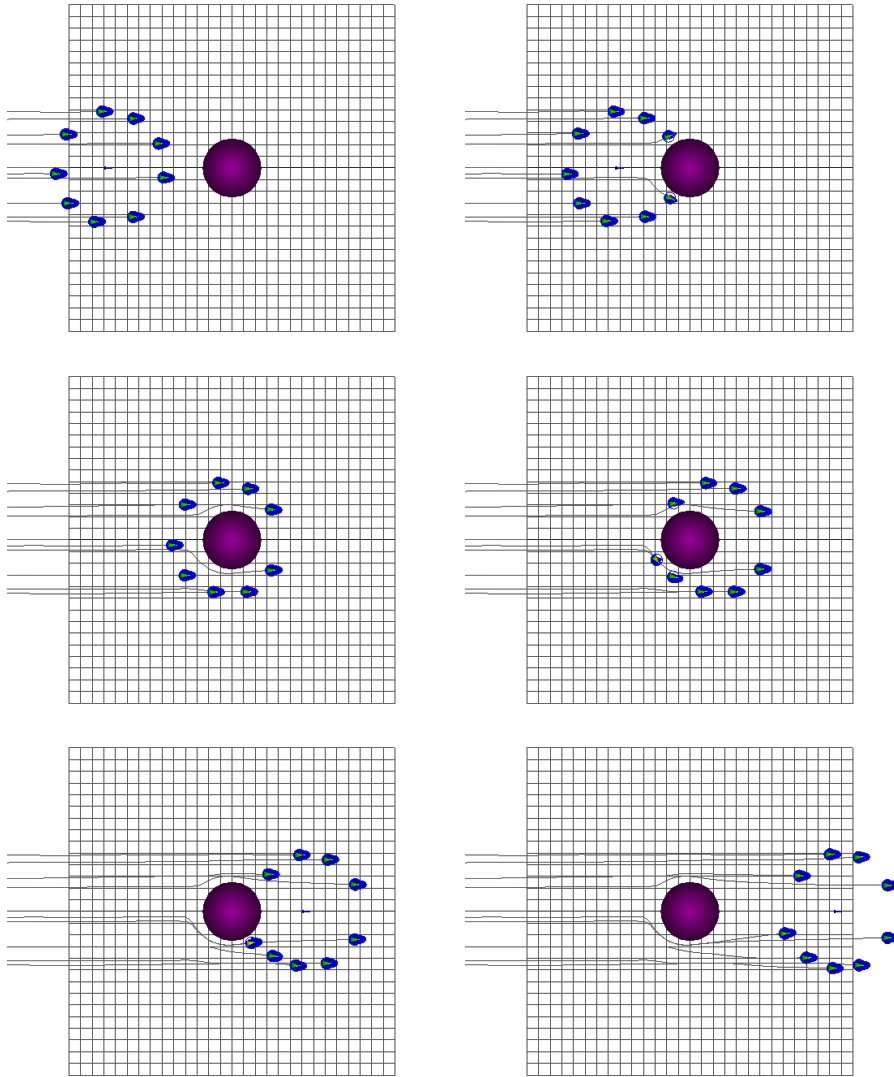


Figure 5.82: Circular formation vs. Column obstacle with 9 robots

Virtual Structures

Virtual structures allow the distribution of the robots in to any desired shape. The main disadvantage of a virtual structure is that, by setting a fixed place for each robot in the formation, the formation is not easily scalable to fit any number of robots, unless it shows a regular o logic pattern to automatically increase the number of spots depending on the number of available robots.

However, it does not require much effort to adapt the already established collection of situation-actions to allow the use of virtual structures. Actually the same situation-action set is used and only the formation field construction is modified.

Figures 5.83 and 5.84, show two different virtual structures: a dense diamond and a smiling face. To achieve these results, at each iteration the robot chooses the most suitable spot in the structure for itself: the nearest one that is empty. When two robots move towards the same spot, the different precedence actions established for the formation-keeping are applied; So one of the robot is able to move into the spot while the other cannot; At the next iteration, the robot that was not allowed to move into the spot chooses a new spot –since the previous one is already occupied– and moves towards it. Using this simple decision mechanism, the formation field is reduced to a single punctual attractor placed over the selected spot in the structure, and the robot navigates according to that field.

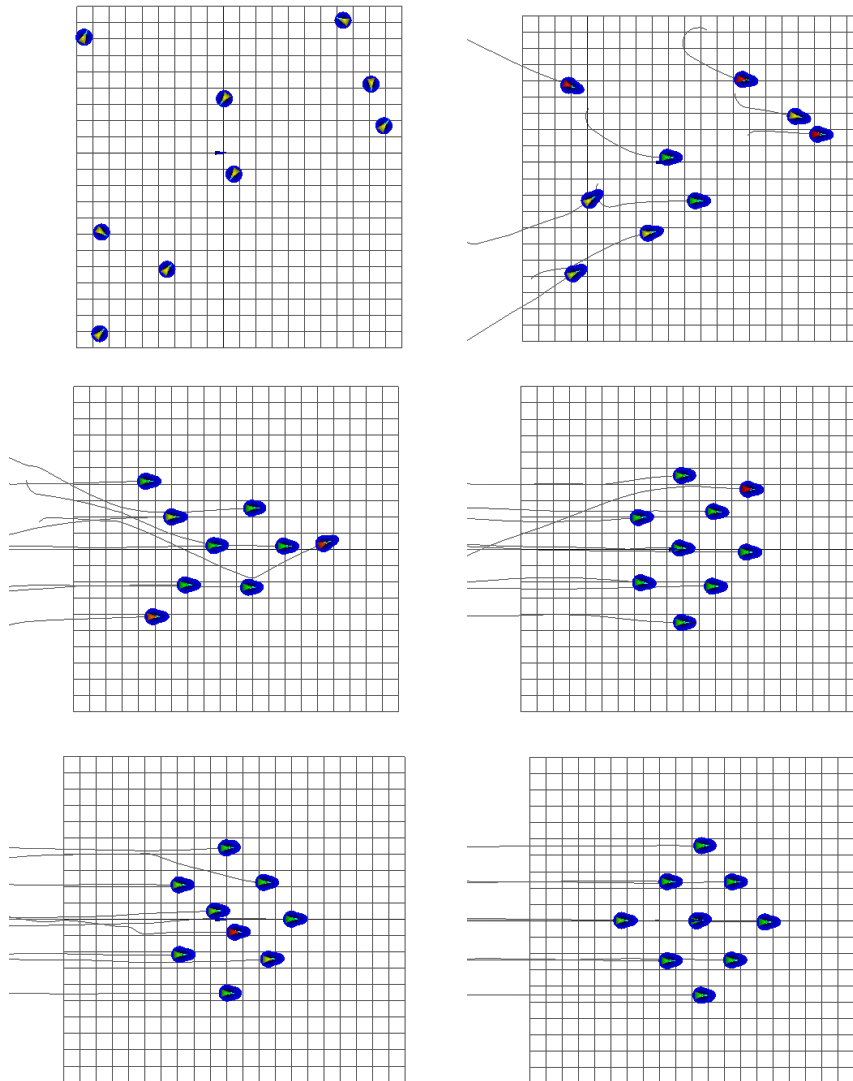


Figure 5.83: Diamond virtual structure with 9 robots

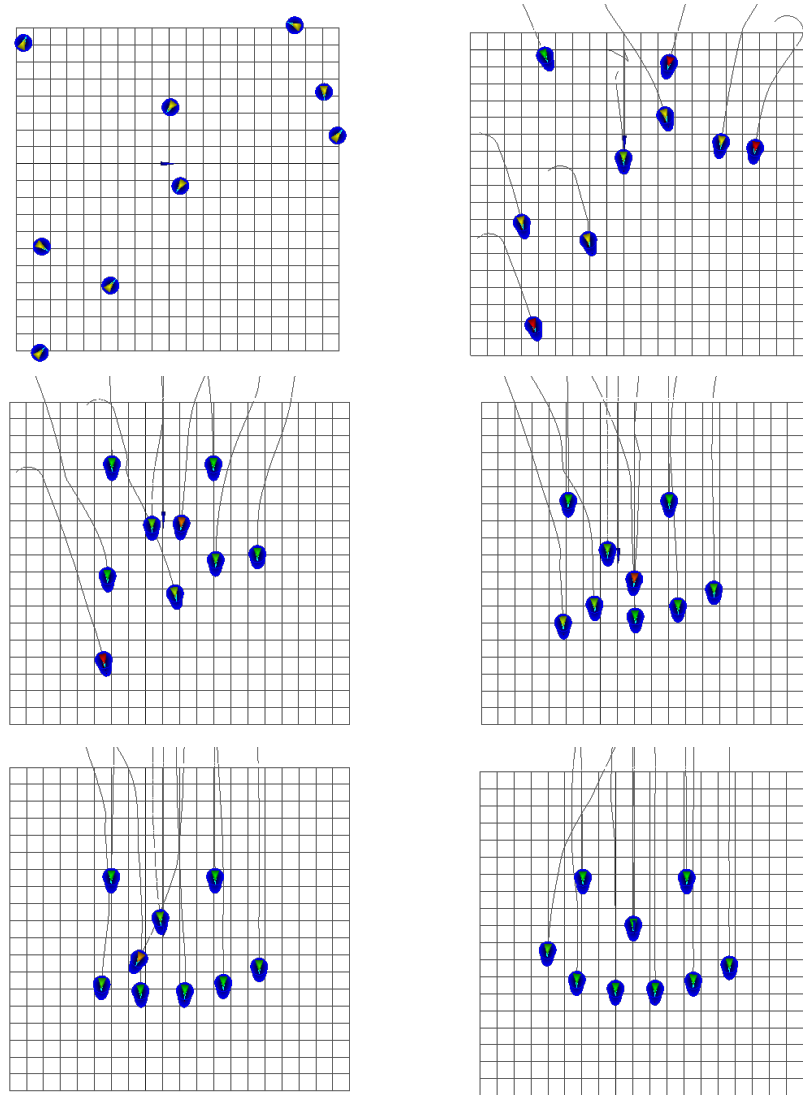


Figure 5.84: Smiling face virtual structure with 9 robots

Chapter 6

Real Experiments

6.1 Overview

In order to test the method in a real environment, a group of robots have been built –more details on the appendix– and the proposed method has been tested using them.

As the experimental space and the number of real robots are limited, it has only been possible to test formations with 3, 4, 5 and 6 robots.

Two ceiling cameras are used to track the robots along the test. The images obtained from them, superimposed with the formation shape and individual robot traces, are shown in figures 6.1 to 6.16. The position of the formation shape and the robot traces, are not fully aligned with the actual formation images because the images were taken asynchronously and the exact timing of each frame was not recorded. However the formation shape provides a good enough reference to appreciate the positioning of the robots.

The first real test can be seen at figure 6.1, where three real robots build a line formation. The robots are initially placed randomly at one side of the experimental space and they evolve to the formation line along the shown frames.

The oscillations appreciated at the beginning were due to the collision avoidance mechanism between the two robots at the back of the formation. This oscillations end when one of them gets near the formation while the other one brakes in order to provide space.

It can be seen in the third frame the effect of the action introduced in table 5.12 to set preference for robots at one side of the line over the other ones. The last robot in the formation has slowed down and remains at some distance at the side of the formation until all other robots pass ahead. In the fourth frame, when there is space for it to incorporate to the formation, the robot at the side gets close to the formation line.

At the end, it can be seen how all three robots fit in the formation line leaving a safety space between them.

At figures 6.2 to 6.16, similar tests are shown for the different formation shapes with 3 to 6 robots.

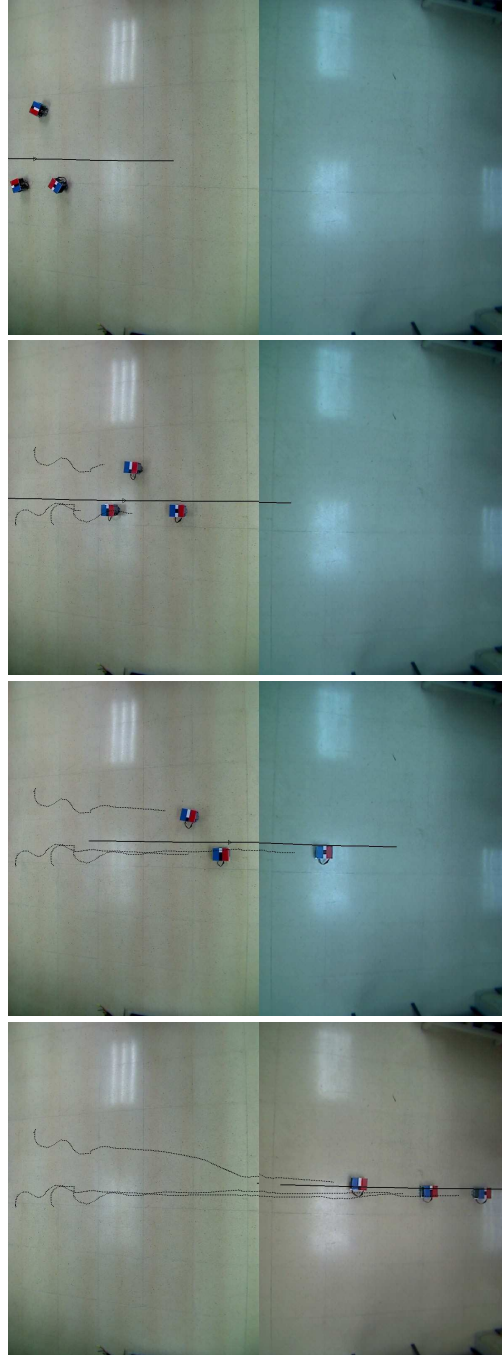


Figure 6.1: Line formation with 3 robots

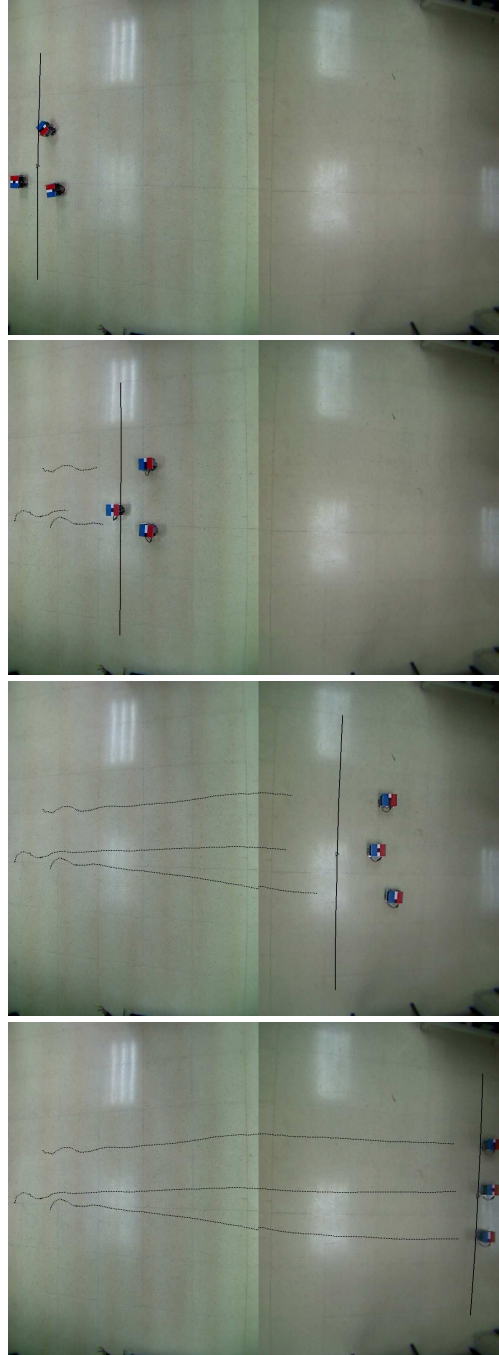


Figure 6.2: Column formation with 3 robots

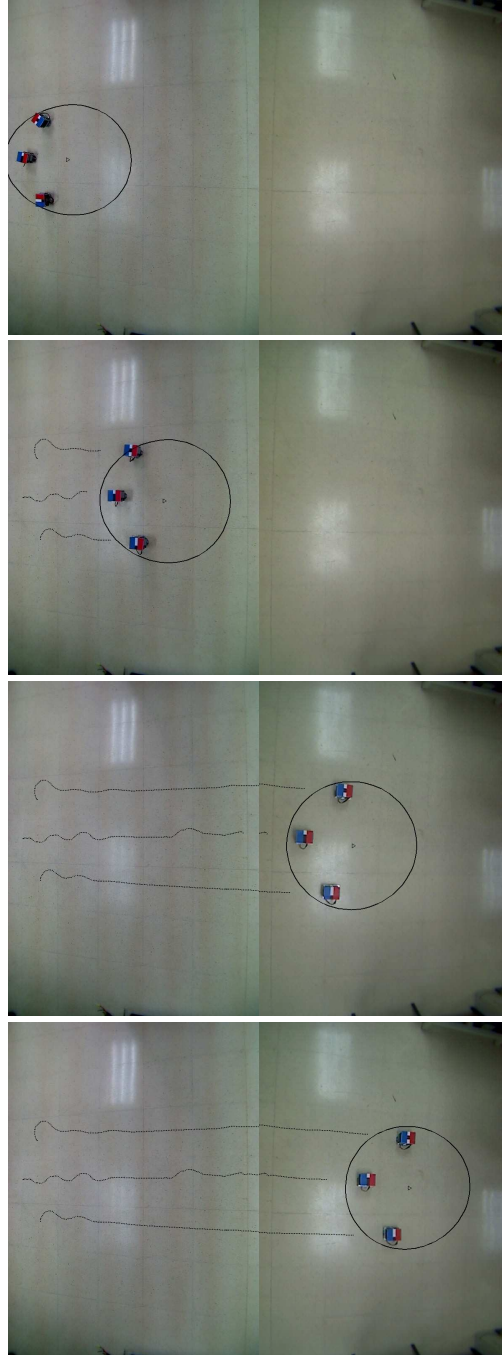


Figure 6.3: Circle formation with 3 robots

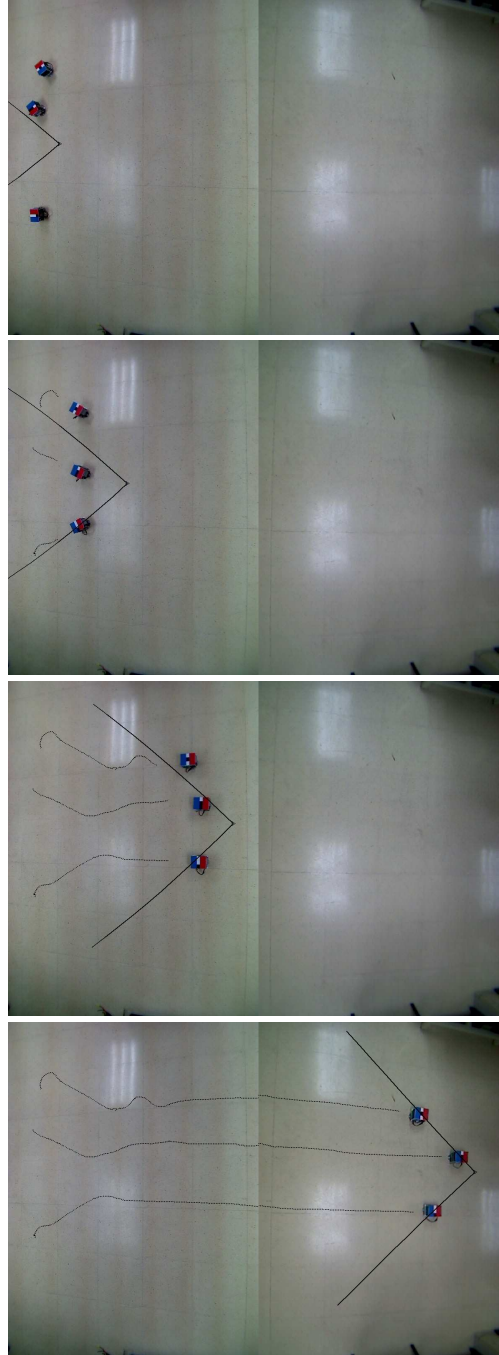


Figure 6.4: Wedge formation with 3 robots

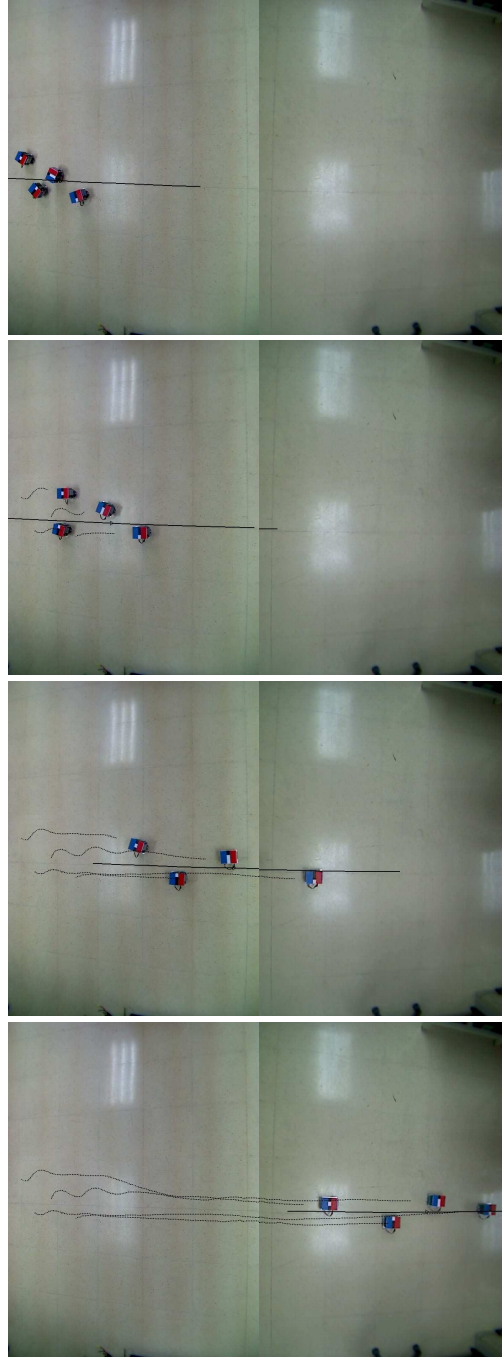


Figure 6.5: Line formation with 4 robots

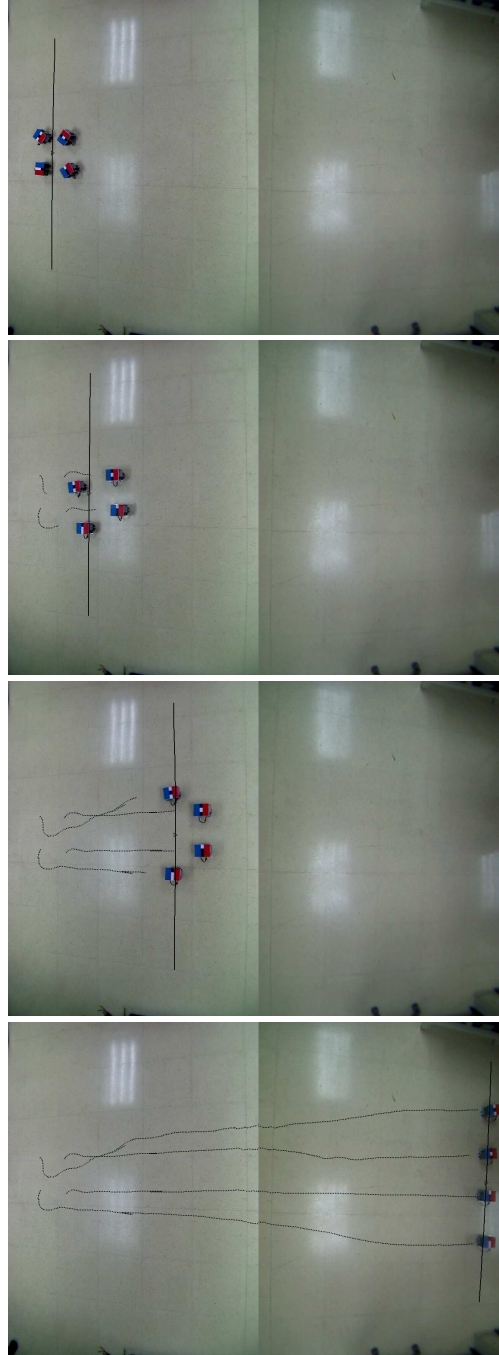


Figure 6.6: Column formation with 4 robots

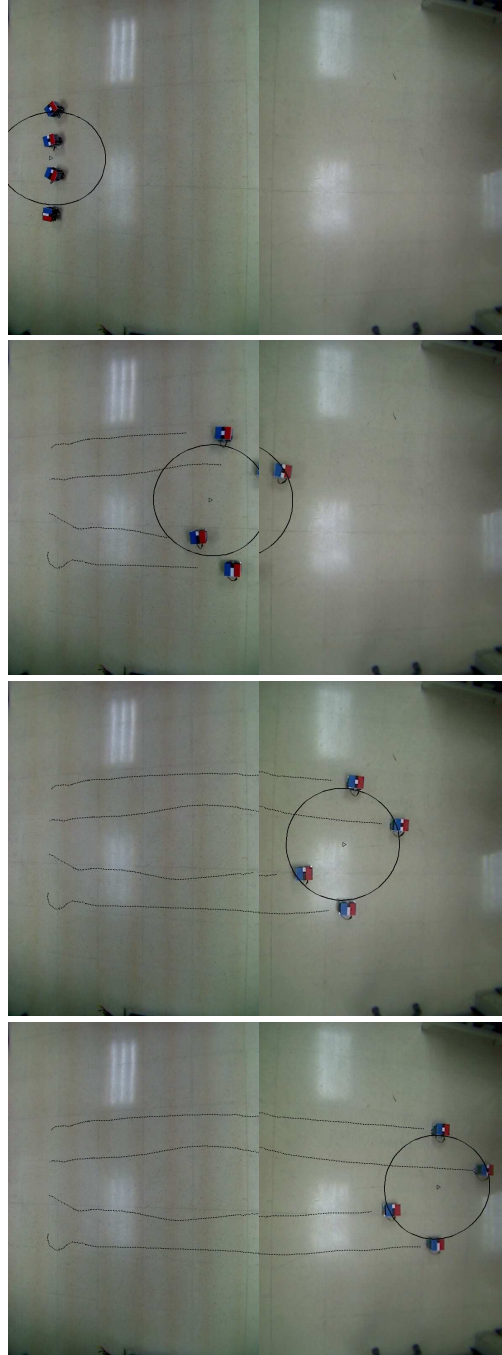


Figure 6.7: Circle formation with 4 robots

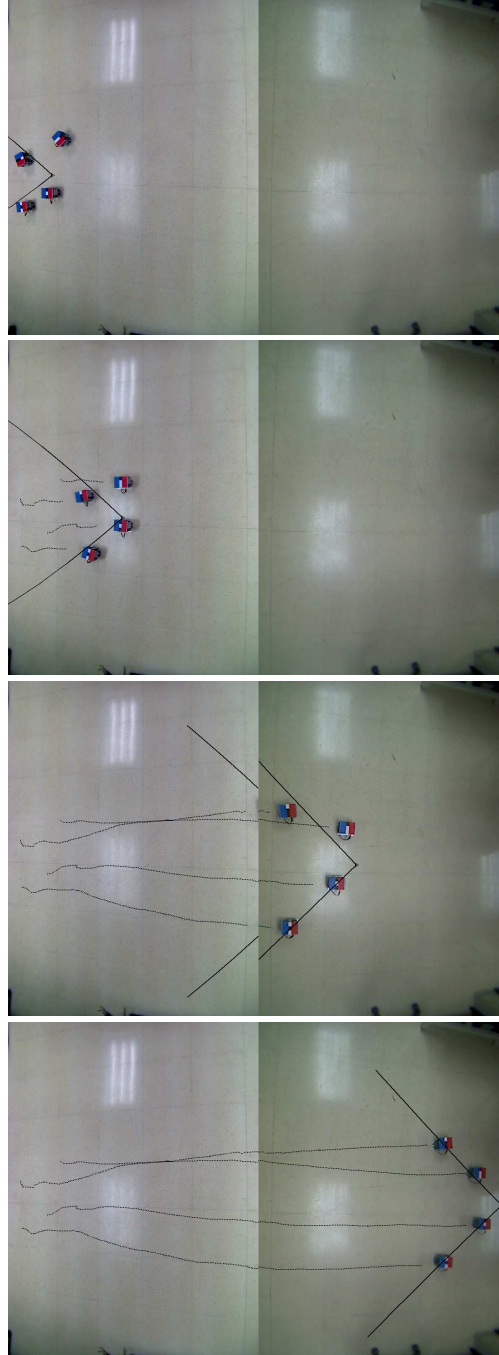


Figure 6.8: Wedge formation with 4 robots

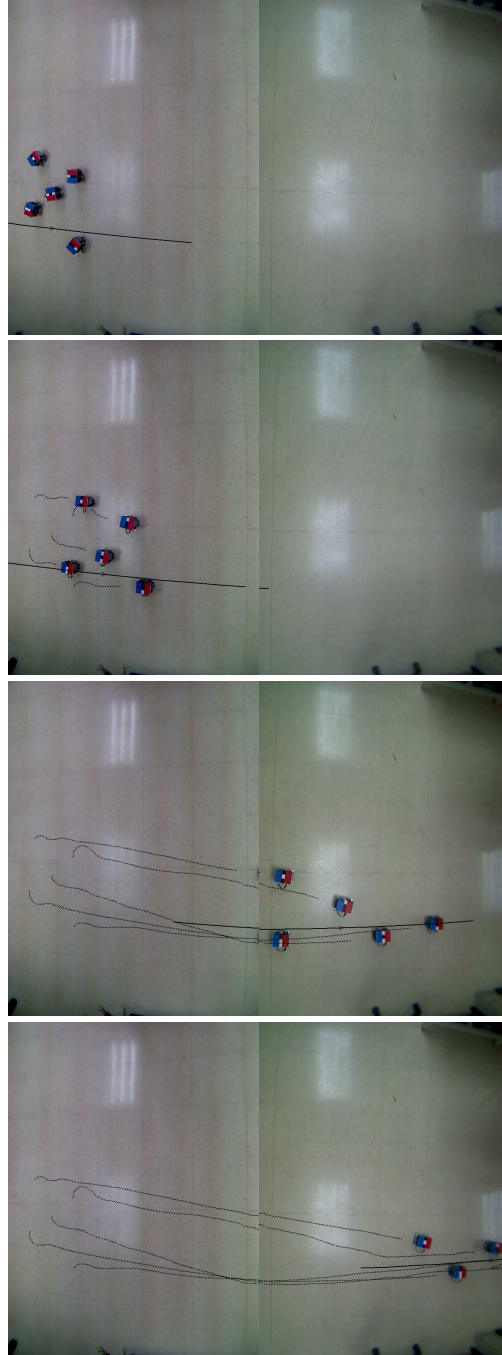


Figure 6.9: Line formation with 5 robots

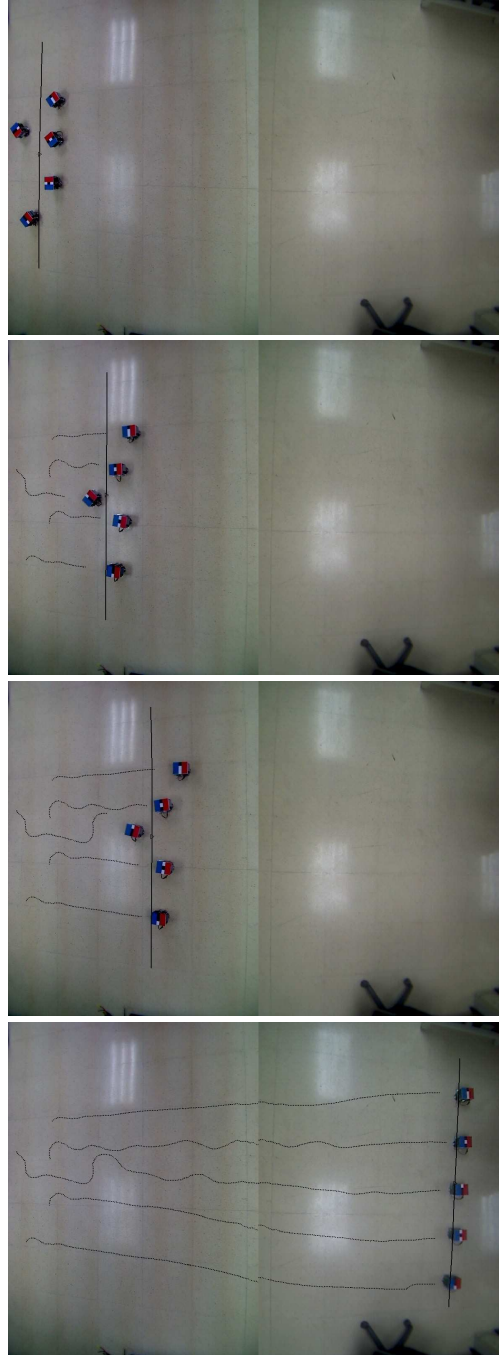


Figure 6.10: Column formation with 5 robots

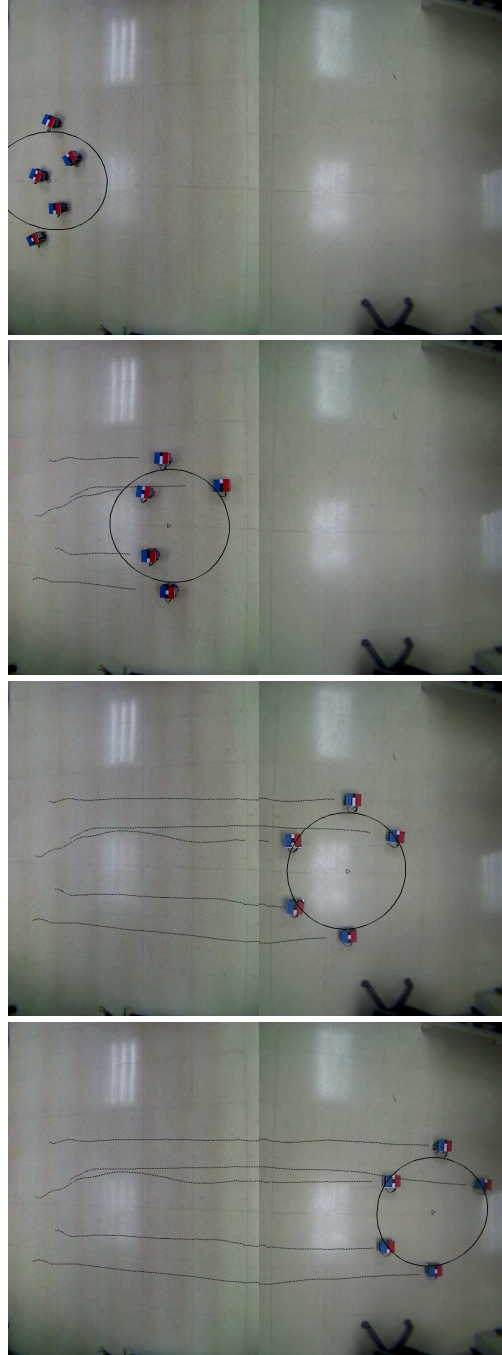


Figure 6.11: Circle formation with 5 robots

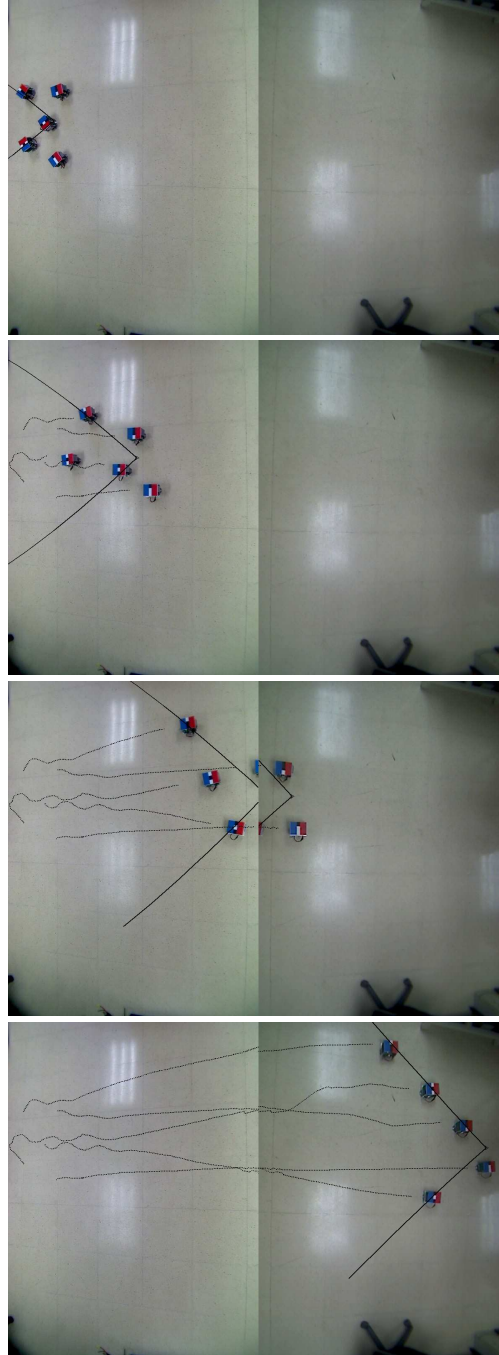


Figure 6.12: Wedge formation with 5 robots

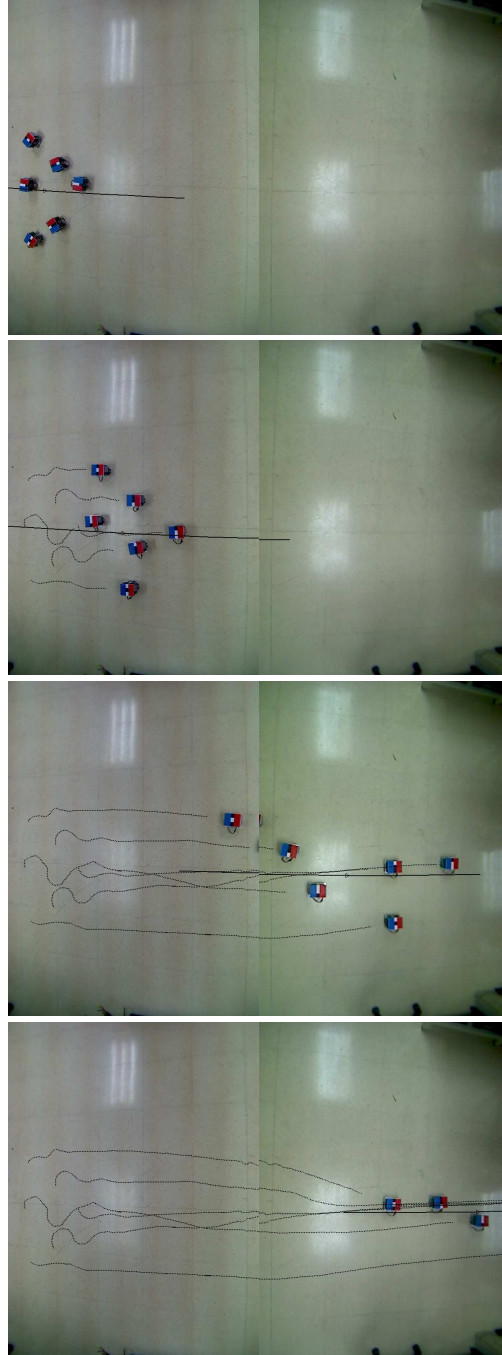


Figure 6.13: Line formation with 6 robots

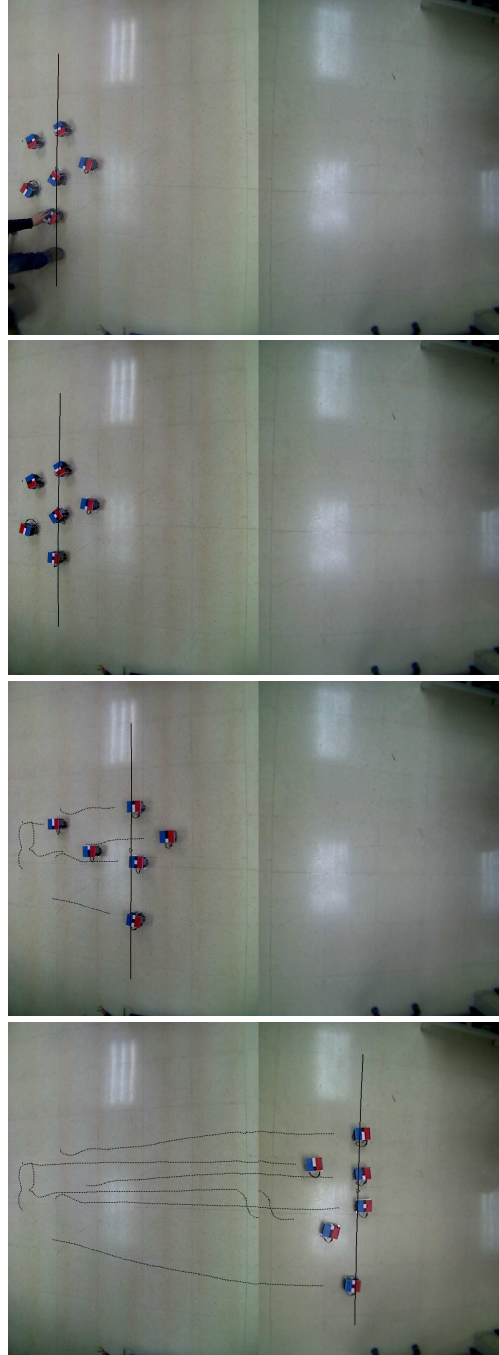


Figure 6.14: Column formation with 6 robots

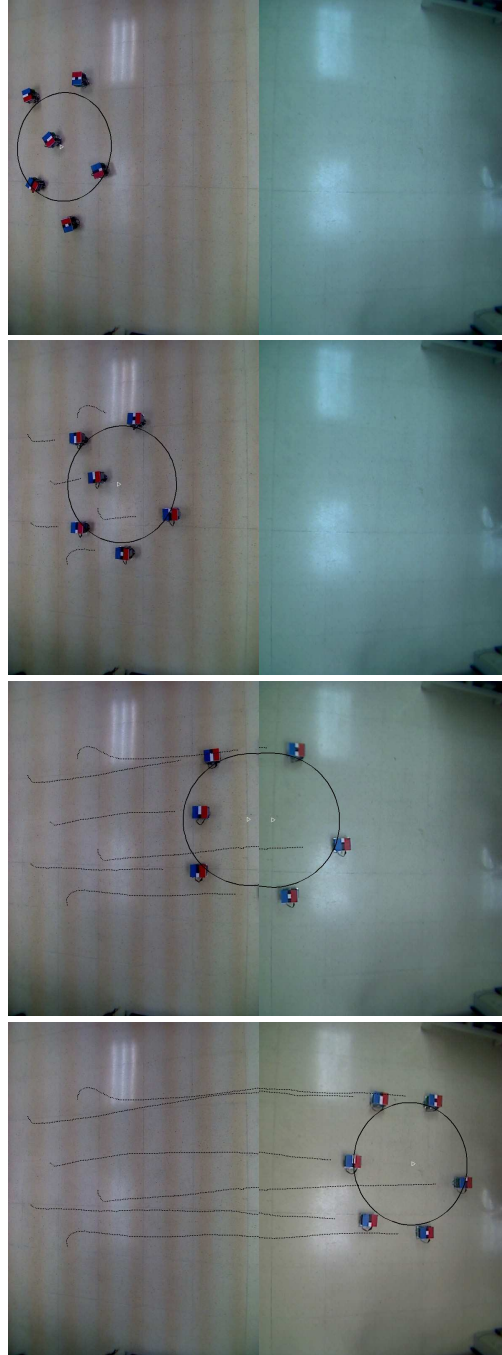


Figure 6.15: Circle formation with 6 robots

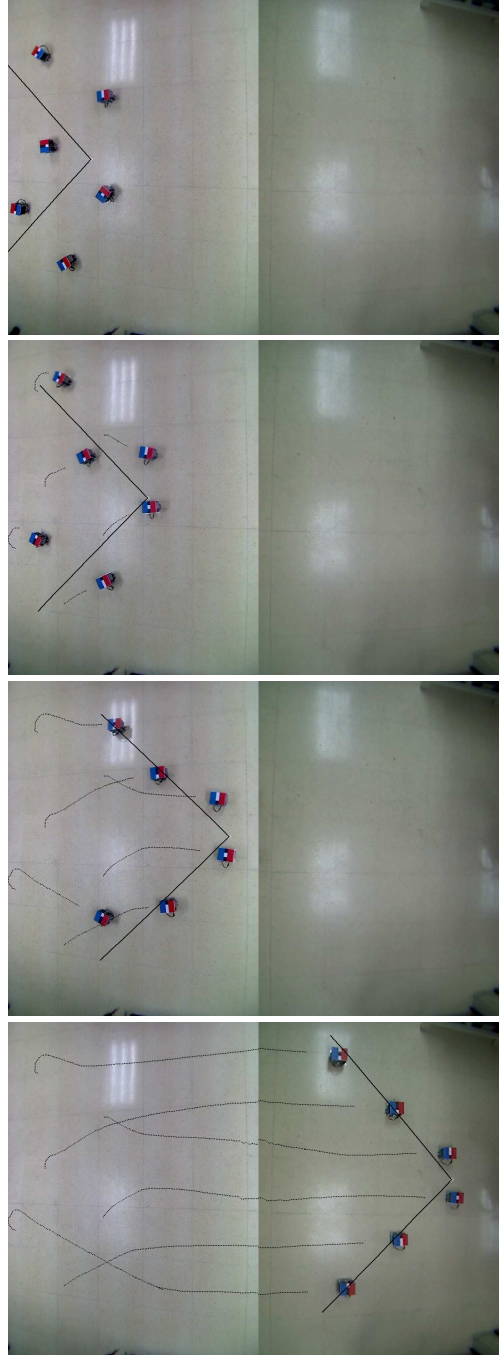


Figure 6.16: Wedge formation with 6 robots

6.2 Statistical Analysis

While the experimental images provide a good overview of the real experiments and reveal a behaviour similar enough to that observed in simulation, to properly compare the results of the real setup with those obtained through simulation a systematic repetition of the real experiments is needed to extract similar statistical observations.

Due to the limited real experimental space available the systematic repetition of experiments has been limited to groups of four robots. It is possible to arrange the formations with more than four robots, but only when using some initial configurations –placed close to the edge of the experimental area–, thus preventing proper random initial positions.

The four robots formations were chosen because they allow the formation gathering within the experimental space with greater variety of initial configurations.

The line formation shape has been also dismissed from the data analysis due to its results. Half of the tested initial configurations were not able to gather within experimental region with all the four robots inside the images at the same time. The robots leave the experimental region through the same spot, but this was not considered enough to be used as a valid result.

Up to 40 repetitions for the column formation and 20 for the circle and wedge formation shapes have been done using four robots. The results have been analysed in the same way as the simulation data: building the curves of distance to formation along experimental time and the arrangement time histogram for each one of the formation schemes. These plots can be seen in figures 6.17 and 6.18.

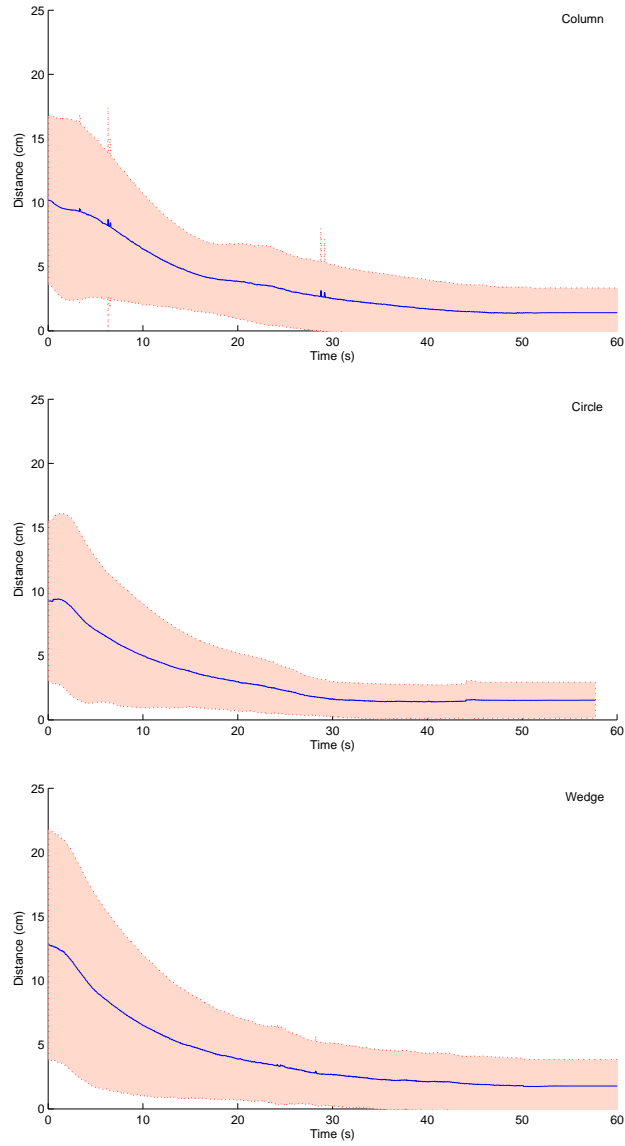


Figure 6.17: Experimental mean distance to formation and its standard deviation for 4 robots

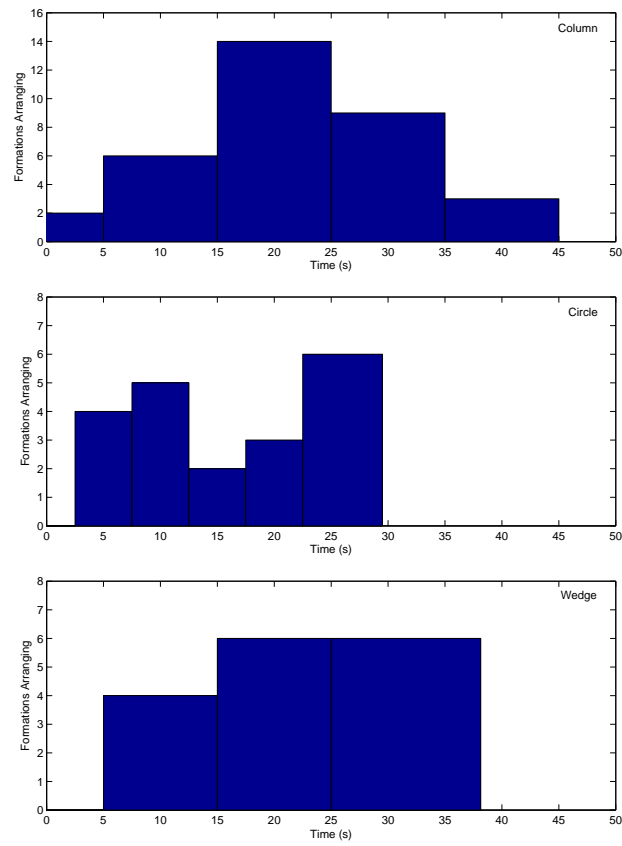


Figure 6.18: Experimental Arrangement Times for four robots

In order to compare this results with the simulated ones, the distance to formation and its standard deviation have been fitted to the same exponential function shown in equation 5.55. The fitting coefficients can be seen at figures 6.19 and 6.20 placed beside the same coefficients obtained for the noisy simulations.

It can be seen how the fitting results of the real experiments are quite similar to those obtained through simulation, but not equal. This is probably due to some real conditions of the experiments not considered in the simulation. The main error source observed in the real experiments is the slippery of the wheels on the floor, which was quite notable –the noisy simulations were done using a proportional noise in the actuators, the slippery of the floor adds a random systematic dragging error factor–.

The speed control of the real robot was based solely on the local measurement of wheel movement, not using the global positioning system for this –to avoid a second control loop interaction with the method–. The result was that the robots do stablish the desired speed but they sometimes move actually at a slower speed for small periods of time.

This kind of problems is reflected on the data analysis, especially in the decay rate. Their effects are barely appreciable when looking at the experiment itself, because the formation positioning part of the method –which uses global positioning– adapts to these external events and keeps the formation in shape. Again, this ability to adapt to external events is the key and the main strength of the reactive methods.

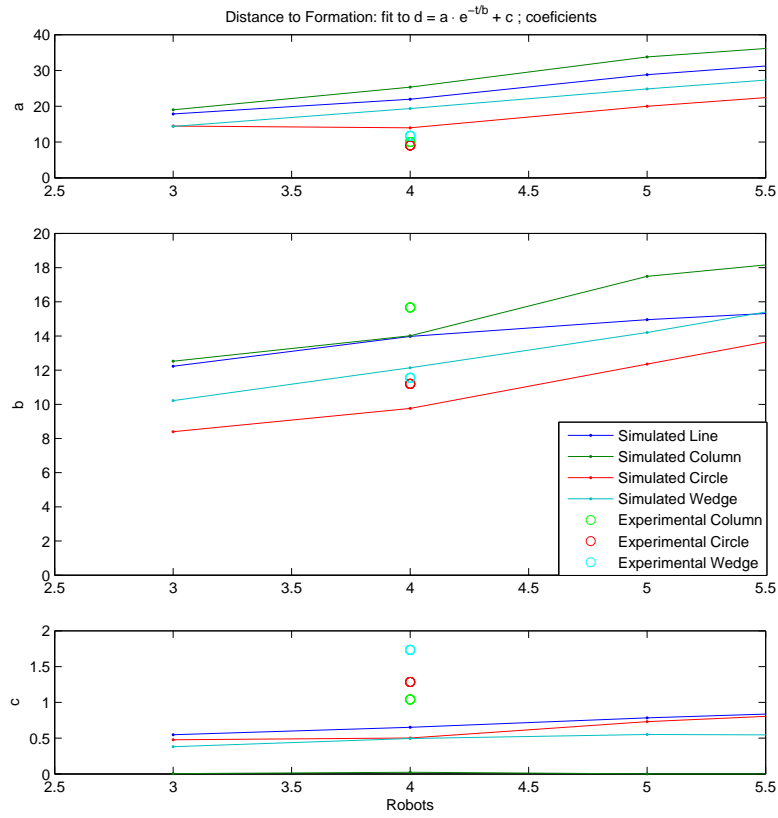


Figure 6.19: Experimental mean distance to formation fitting coefficients against coefficients obtained at the noisy simulations

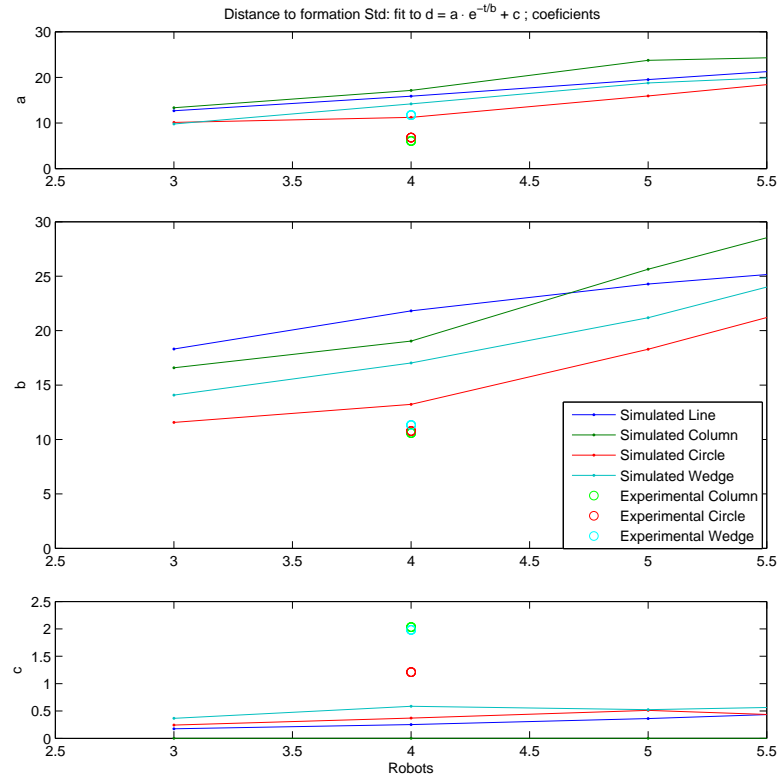


Figure 6.20: Experimental mean distance to formation deviation fitting coefficients against coefficients obtained at the noisy simulations

Chapter 7

Conclusions

Along this work it has been established how the movement instructions to be followed by a robot can be expressed in a way –richer than just providing the desired movement direction– that ease the resolution of conflicts due to the situation analysis of different aspects of the robot environment.

While the actions used along this work have been oriented to handle the robot navigation, the same principle can be applied to other aspects of robotic design. Potential conflicts can be easily avoided by establishing an action space with higher dimensionality than the actuator space, where the extra dimensions of the action space are oriented to disentangle the potential conflicting commands.

Along with the new way to state the robot navigation actions, the work proposes a method to merge the different actions –originated by the analysis of different aspects of robot situation– into the actuation space.

The merging method proposed along this work is mainly aimed to non-holonomic robots, like those used for the simulations and real experiments. However, the different aspects involved in the construction of the merging algorithm have been isolated to allow an easy adaptation to other kind of robots.

In order to test the proposed situation-action description and its merging method, a collection of scenarios with increasing complexity has been proposed and solved. The proposed scenarios have shown how the proposed situation-action description can be systematically applied and how this methodology can benefit from re-usability of situation-action sets as the complexity of the scenario increases.

The first solved scenario, a single robot with an obstacle in the path to the goal, illustrates the basics of the situation-action description and use. This scenario has also shown how the most common problem of the potential field approaches, the local minima, is avoided by the proposed method.

The second scenario, simultaneous navigation of multiple unrelated robots, shows how the basic situation-action set developed for the single robot scenario can be easily expanded to handle the coexistence of multiple robots.

The third proposed scenario, where multiple robots behave as a group but without any given structure, shows the main differences among the navigation of individual robots and the robots acting as a group. Again, this scenario benefits from the easy adaptation of the situation-action sets employed in the previous scenarios.

The last studied scenario, formation navigation of a group of robots, adds and structured relationship between the robots of the group while keeping all characteristics developed for the previous scenarios.

In this last scenario a more extensive study of the performance of the method and the repeatability of the results has been done, both through simulation and with real robots.

In the analysis of the results, a clear linear tendency with the number of robots is obtained. This linear tendency supports the idea that the reactive scheme proposed for this work is able to handle any number of robots in the scenario. This is done by the distribution of the complexity of the complete scenario among the robots, where a centralized control should deal with a complexity that grows geometrically with the amount of robots.

Also, the dispersion of the obtained results indicates that the proposed method can be employed in most of the situations, and their results will remain consistent independently of the initial conditions of the robots.

The validation of the simulation process is also done in this last studied scenario. The application of the proposed methods to guide real robots have been shown, and the movement observed on the real robots navigation was comparable to the movement obtained in simulation. The results on the performance tests obtained using real robots was comparable to the results obtained through simulation. The slight deviations obtained was small enough and can be explained by some factors not considered in the simulation.

Application and Future work The most immediate applications for the developed method are in the deployment of teams of unmanned vehicles, and in structure keeping of multi-robot tasks.

While the main tasks covered along this Thesis have been the navigation of the robot and the coherent navigation of a group of robots, the used method can be applied to other areas of robotics.

In general, applications that uses multiple robots, will benefit for the application of this work due to its capability to handle, in the same way, those aspects related to the individual task and those related to the team task.

Without moving far from the navigation, tasks that must handle multiple robots and evaluate multiple aspects of the robot environment in order to provide proper results, like robot soccer or other team-oriented tasks, can highly benefit for the proposed method since it allows an easy way to merge multiple considerations on the robot movement.

Another possible application of the proposed method are in the integration of heterogeneous robot groups. The current research uses homogeneous robots, but, since the dependence of the behaviour on the robot characteristics has been reduced to the minimal, it is reasonable to expect that heterogeneous robots can be used. However this will require the study of which characteristics belong to the group and which ones are only dependent on the individual.

It will be also possible to improve and generalize the proposed method by the combination of the actual world modelling, actually limited to the virtual potential fields, with other models focussed on some especial aspects of the action maps –like the use of the original Vector Field Histograms to enhance the forbidden action map–.

Appendix A

Simulation Environment

The simulations used along this work were done in a custom simulation environment.

At the time when this research was initiated some robot simulation engines were available, but none of them seems suitable due to stability, maintenance, or lack of capacity for including the custom elements needed for the task.

When the simulation environment was firstly conceived, some experience with building and programming real robots has been acquired. From this experience, the simulation environment was designed to work in a way that mimics how the software in the robot is handled and the kind of data accessible to the robot.

Also, facing a project where the number of robots in a given scenario is variable and where multiple scenarios were going to be tested, the simulation environment was built in a modular fashion so both the devices tested and the scenario itself could be easily modified.

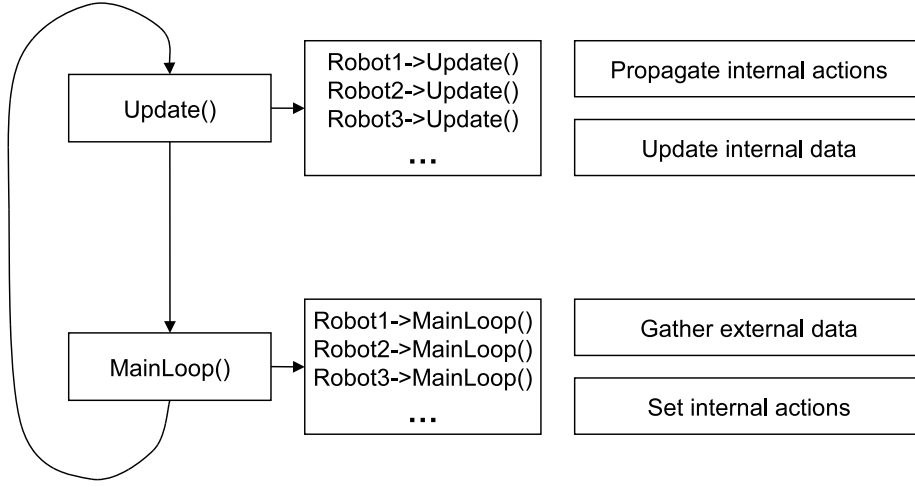
The developed simulation software is divided in different modules for the simulation process: the visualization of the scenario, the properties interface, the information interface, the simulated robots and any other entity placed in the scenario is built as an independent module. The main simulation program just provides the essential elements to load the different modules, control the simulation process and coordinate the data flow in the scenario.

A.1 Main Elements

Any module being used in a scenario is loaded by the main simulation program and then integrated within the simulation through a collection of common methods and data collections. The elements that are simulated, such as the robots or the obstacles, are referred to as *Devices*.

A.1.1 Simulation process

The simulation process is supported by two main methods that exist in any simulated *device*: the `Update()` method and the `MainLoop()` method.

Figure A.1: Simulation Loop for sequential *devices*

These two methods are used to synchronize the simulation execution and keep isolated the actualization of the internal data of each robot and the decision process that uses data from other robots being also simulated.

At the `Update()` method the robot updates its internal values such as position or orientation to get them ready to be used by other robots.

At the `MainLoop()` method, the data from other *devices* in the simulation are gathered. and the values that depend on other robots are updated, such as sensor readings or other robot positions. Once all the data, internal and external are updated, the decision process to establish the next actions of the robots can be done.

The *devices* used in an scenario can be processed both sequentially or in parallel, depending on the scenario configuration, however, only the information published at the end of the `Update()` method is accessible for other robots. For those *devices* processed sequentially, all the `Update()` methods are executed together and then, all the `MainLoop()` methods are evaluated as it is shown in figure A.1

A.1.2 Information structures

Each *device* inserted in the simulation is identified through its *Name*, which is unique for each device, and *Type*, which allows to identify devices built from a same module –Every robot has a different name, "Robot 1", "Robot 2", ... however all the robots have the same type "Field Robot"–.

The information shared by a *device* with the other elements in the simulation is contained within four lists: the *Block List*, the *Argument List*, the *Info List* and the *Actions List*. Every *device* has access to the list of all the *devices* present on the simulation, and acceding to a specific device from the devices list, provides access to the four lists of that device.

Each of these lists contains different aspects of the *device*; each list contains a set of elements that, together, completely describe the robot and its current

state. Each device is responsible of populating its own list in order to interact with the user and with other devices.

- The *Block List* just provides a quick access to the list of robot characteristics. When the Block is public, it allows others to modify its content; this modification will be applied in the data contained in the Block owner.

Each Block is a group of common properties and methods that are related between them and usually need to be evaluated together, like x,y,z in position, the three orientation angles or width length and height for body size.

The Blocks contained within an element are considered permanent along the simulation; the values of the properties contained within the block will vary along the simulation, but the properties will be always present.

The Blocks provide a functional description of the *device* itself and support the possible interactions between *devices*.

- The *Argument List* is a collection of particular properties of the *device* which can be modified by the user at the beginning or along the simulation.

The Arguments are the kind of information that will characterize the robot behaviour for a specific scenario: control constants, boolean switches enabling specific behaviours or specific elements for the current scenario like the kind of field.

When an Argument is changed, an `OnChange()` method is called for that argument, so specific actions could be taken in the device to reflect the change of the value, even the rejection of the change.

The information contained in the *Arguments List* will be stored to a file when a simulation scenario is saved, to be launched again with the same conditions. The Argument items are constant along the simulation for each *device*.

- The *Info List* is similar to the *Argument List* but it is intended to hold status values that are useful to the user to follow the device evolution along the simulation. The values at the *Info List* cannot be modified by the user or any external *device*.

Each Info item contains a particular property to be observed by the user, monitored by other devices or logged for later analysis. The Info items are accessed by name but differ from the Blocks and Arguments because the list of Info items can be modified along the simulation.

- The Action List contains a collection of methods that can be triggered from outside of the device, usually by the user, to perform some predefined actions related with the internal state of the device, usually not directly accessible from the outside. This can include to reset some internal counter, create a snapshot of internal values on a file or just toggle a specific robot behaviour.

Time control on the simulation is done through the main simulation program. Fixed-time step and proportional-to-real-time step are available as time flow controllers, however only step ahead is considered as time control strategy;

variable time step with step back is not considered. The proportional-to-real-time step can be used to evaluate the algorithms performance with the the algorithm computation time effects on the robots behaviour; it has been also used to run scenarios with a mix of simulated and real elements.

A.2 User Interface

The main simulation program provides a very basic interface, just a collection of menus to load the different modules or scenarios, to select the elements at the simulation, and to control de simulation process.

The interface modules follow the same philosophy of modularity used in the simulation devices. Each interface element is built as an independent module, that is added to the simulation main program when needed. In this way, different visualizations can be easily deployed for each scenario and different elements can be added without needing to modify the main program.

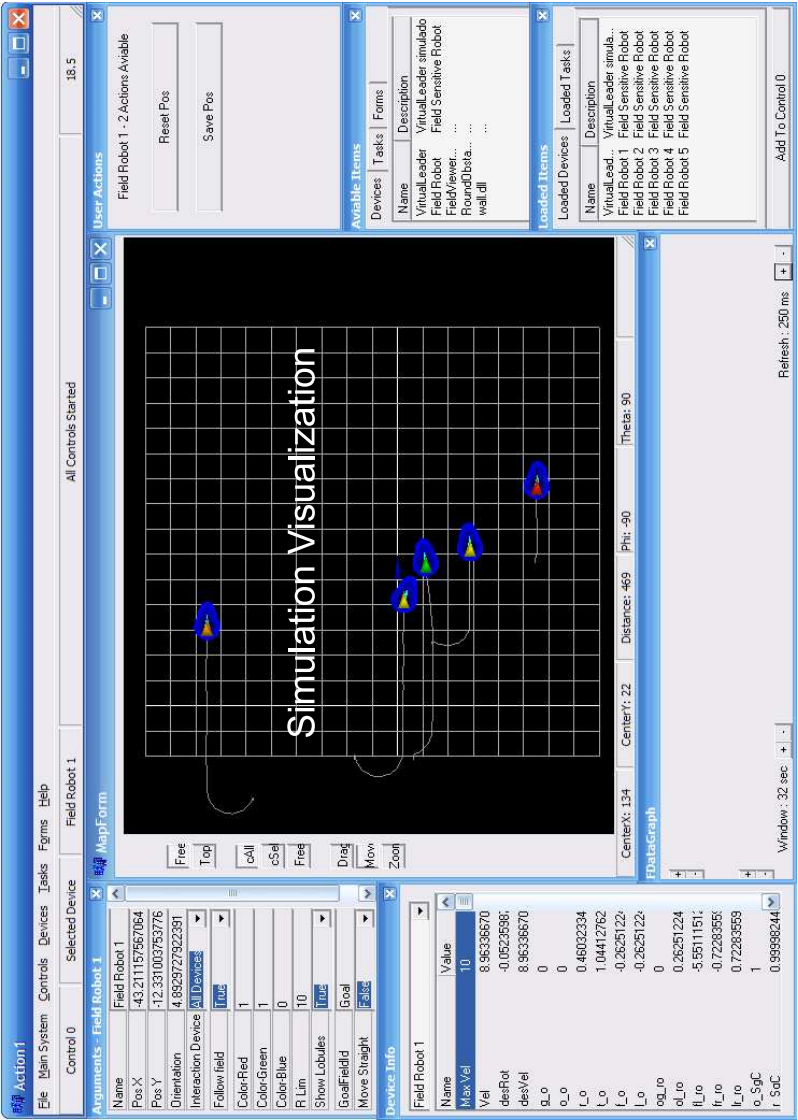
The user interface runs mainly independent of the simulation execution. It has access to the devices included in the simulation by synchronizing the interface data with the robot data after the `Update()` and `MainLoop()` methods through the lists shared by the devices.

The main interface elements are the *Argument List* visualization, the *Information List* visualization, the *Actions List* launcher and the visual representation of the simulated devices. Other interface modules such as the list of loaded devices and the *Block List* visualization are also usually loaded for a default simulation scenario. The representation of the simulation evolution of the scenario is done through OpenGL and a particular *Shape Block* which contains the 3D figure of the specific device.

Device Information

Device Configuration

Main Window



Devices in Scenario

Device Actions

Figure A.2: Simulation environment visual interface

Appendix B

Experimental setup

To validate the simulation and test the proposed method in real robots, a group of small robots has been built. Since the test scenarios involve the use of multiple robots, the built robots have been kept small enough to allow the deployment of the real scenarios in a reasonable space.

The robots have been designed to operate in a *remote brain* configuration.

The robot controller takes care of low-level tasks, such as the motor control or sensor data recollection, and mid-level tasks, such as speed control or orientation keeping.

Along with this, a mirror image of the robot is kept in a remote computer, where a local agent –the representation of the robot in the computer– executes the body of the proposed algorithm.

The robot and the computer are connected through a radio link used to send the monitored data from the robot to the computer and send the high level commands from the computer to the robot.

The data sent from the robot to the computer include the current speed, current position and turn ratio according to local sensors, while the computer sends to the robots the desired speed, desired turn ratio and desired heading according to the proposed method.

This scheme allows the execution of exactly the same navigation algorithm in both the simulations and the real robots tests, and eases the execution traceability in the real robots tests.

To provide absolute positioning to the robots, a set of cenital cameras –on the ceil, pointing to the floor– has been placed to track the absolute robot positions and to correct the local positioning of the robots. This information is processed in the remote computer and available to be used by the agents running the high level navigation algorithm.

The computer agents combines the absolute position obtained through the cameras with the local position provided by the robots to get a better estimated robot position along the execution of the proposed method.

B.1 Robot Hardware

The real robots, shown in image B.1, were built to keep them simple and expandable so they can be used along this work and for further work.

The main mechanical actuators are two stepper motors, directly attached to the wheels, which provide differential drive motion. Over them, a flat platform holds the set of electronics boards which control the robot. The electronics are built around a Microchip PIC18F4550 microcontroller, a MRF24J40MA radio board and two ULN2803 acting as motor drivers.

The robot dimensions, 11.5 cm long x 9.5 cm wide and 13.5 cm high, have been kept reduced enough to allow the simultaneous deployment of multiple robots required for the testing environment. Eight of these robots have been built, however only six of them have been used simultaneously for the testing, leaving the two last ones as backup units.

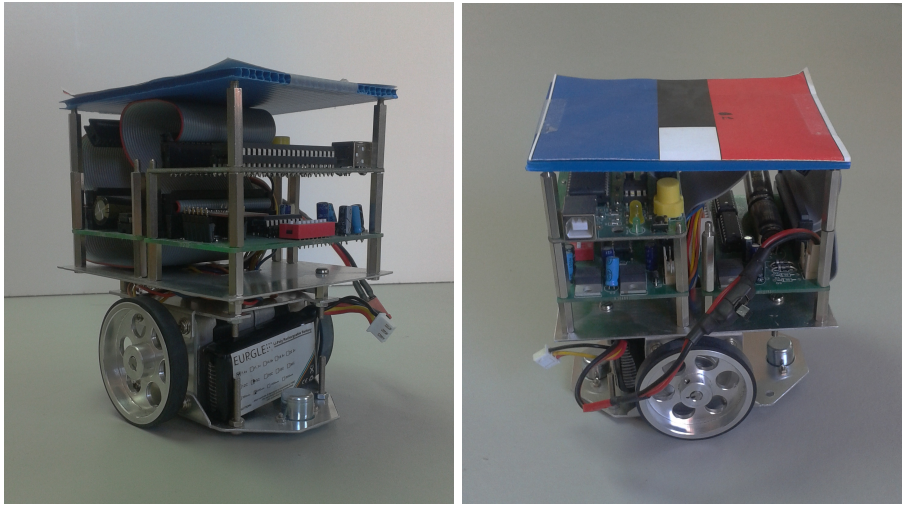


Figure B.1: Experimental Robot

B.1.1 Control Board

The main element in the control board is the microcontroller, which provides the logic system of the robot. This microcontroller runs all the local control algorithms, sensor readings, and radio communications. The used microcontroller, a PIC18F4550 from Microchip, is a High performance 8bit, 48Mhz microcontroller with USB client, SPI, I2C and 10bit ADC as main characteristics.

The microcontroller board, named UsbLab, has been developed jointly with the Computer Architecture and Automatics department of the Physics Faculty at the UCM as a multi-purpose control board, and it is currently used along many projects and student laboratories on both the Physics and Computer Science Faculties. Along this Thesis, a lot of contributions have been done to the development and improvement of the UsbLab hardware and software capabilities.

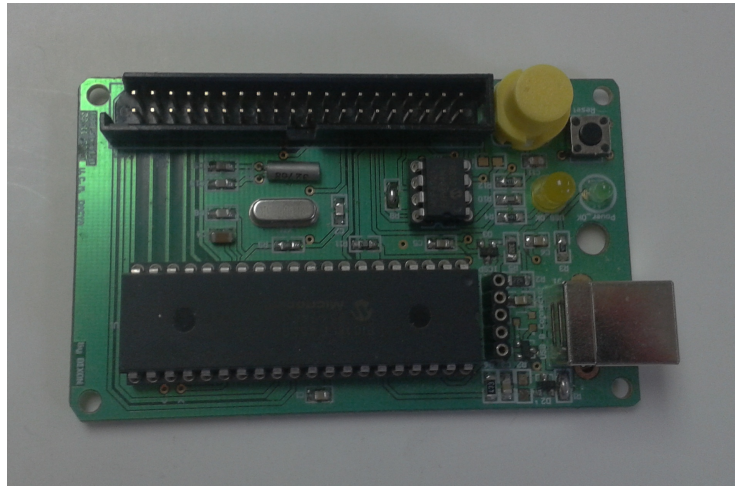


Figure B.2: UsbLab control board

B.1.2 Radio Board

The radio board is built around a radio module from Microchip, MRF24J40MA, and provides wireless communication among the robots and their virtual agents placed at the remote computer.

The radio board hosts, along with the radio module, an 8bit switch used to provide a physical unique id to each robot. Also there are included some power conversion modules to match the radio electronics voltage and power both, the radio and controller boards, through external batteries.

The MRF24J40MA radio module is controlled through a SPI bus and provides Physical and MAC communication layers. Being compliant with the IEEE 802.15.4 standard, it supports Zigbee, MiWi and custom wireless protocols operating in the 2.4GHz band, providing data rates up to 250kbps. While this radio board has been specifically developed for this work, it is currently used in other projects and student laboratories due to its easy integration with the UsbLab.

The same radio board is used in both the robots and the remote computer hosting the virtual agents, with different software in each case.

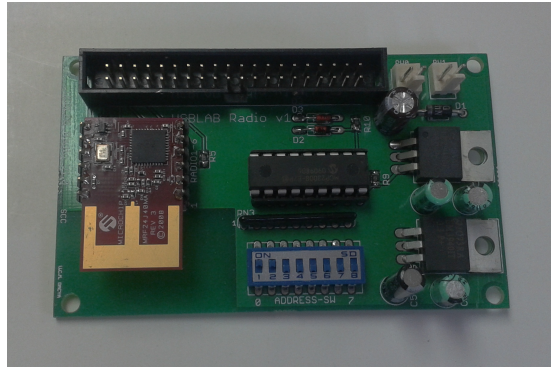


Figure B.3: Radio board

B.1.3 Driver Board

The driver board is aimed to interface low power signals from the microcontroller to high power signals used by the motors.

To drive each one of the two mono polar stepper motors a simple ULN2003 Darlington Array have been used, leaving the coil switching schemes to the microcontroller to smooth the robot movement and reduce the power consumption.

Along the design period a configuration, a dedicated step generation circuit such as LM297+LM298 was tested, however the high torque of the motors and its high power demand causes a very shaky and slippery movement at low/mid speeds, due to the holding of the step current between steps. To avoid these problems the microcontroller, along with the ULN28003, have been used to drive the stepper motors. Providing only small step current pulses and leaving the wheels free to turn between steps. This configuration has shown a smoother wheel movement while drawing quite less energy from the battery .

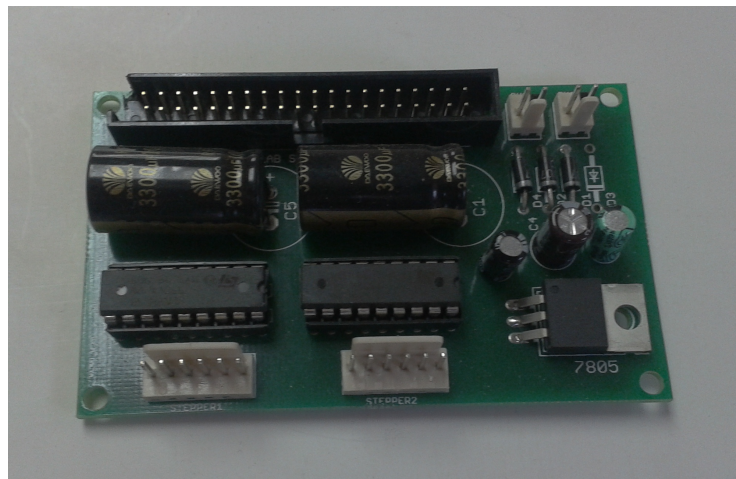


Figure B.4: Motor Driver board

B.2 Robot Software

The real robots software is divided in two parts: one running locally, on the microcontroller, and another one running remotely, on the robot agent located at the computer.

The local software, running on the microcontroller, deals with two main tasks and two auxiliary ones. The two main tasks are to reach and keep a given speed and orientation. Those tasks are fundamental for the method execution, since the navigation algorithm only outputs desired speeds and orientations. The auxiliary tasks executed on the microcontroller are only needed due to the experimental system design. The two auxiliary tasks are: the radio communication with the robot agent on the remote computer and the signal generation needed to drive the stepper motors.

The robot movement control is done through the pulse generation scheme applied to the stepper motors. The studied robots are non-holonomous, while the used real robot are semi-holonomous –they cannot move sideways, but can turn on the spot–. However, by limiting the turn ratio proportionally with the forward speed, the robots act as if they were fully holonomous, with any specified turn ratio. Using the desired forward speed and the limit turn ration the specific speed for each wheel is set for the desired manoeuvre. The pulse generation routine compares the desired speeds for each wheel with the current speed on each wheel and outputs the specific pulse chains to drive the stepper motors accordingly.

The radio link between the agent on the remote computer and the robot is used to send local state of the robot to the agent and, to get the desired orientation and speed commands from the agent. Each robot routinely, ≈ 10 times per second, provides to its agent the current stepper count on each wheel and the local time, in microcontroller tics. These data are used to stablish the robot position in the scenario, once combined with the visual positioning. Knowing the robot position, the navigation algorithm can be executed to obtain the newly desired orientation and direction which are passed back to the robot.

Also a supervision on the robot drift is keep, and corrections to the local orientation are applied to match it with the improved orientation obtained from the combination of step count and visual tracking. Average forward speed is also controlled, however, the spurious slips of the wheels on the floor cannot be compensated due to the lag between the slip and its detection.

B.3 Robot positioning

The final positioning for the experimental robots is done through a combination of local positioning –using local encoders– and external positioning –using cenital cameras–. The combination of these two methods provides an accurate positioning, with small enough errors that will not be achievable using any of the two methods independently.

A common difficulty related with robot positioning is that each individual method has strong points, but also has strong limitations.

While local methods, such as encoder tracking, provides very precise motion

tracking the absolute position error grows along time, rendering the measurements useless after a small amount of time.

On the other hand, absolute positioning methods, work in a way that their precision does not vary along time. However the precision decreases, or the computation time needed to get the data increases, as the area to cover grows.

B.3.1 Local Positioning

In the case of the experimental set-up used for this Thesis, the local encoders provide readings with a movement precision around $\approx 0.05\text{cm}$, but these readings become unreliable after some tens of centimetres; even less when there are many turning manoeuvres.

The main source of errors when using local encoders to track position is due to the wheels slipping on the floor, so the wheel movement does not fully represent the robot movement, and since encoder tracking works just by counting the wheel movement, the errors due to slipping accumulate over time.

The local encoder information is obtained at the robot and must be transmitted to the remote computer, where the final positioning is being solved by the remote agents. This transfer, since it involves very little information, can be done very quickly. A refresh rate of 50Hz has proven good enough without demanding much effort from the robot or the remote computer.

B.3.2 Visual Positioning

Global positioning is here solved using cenital cameras. In these kind of systems the accuracy of the positioning is directly related with the resolution used for the image captures and the area covered by the camera. The main advantage is that the positioning error is mainly constant along all the covered area –it only varies due to the camera perspective–. However, to achieve high positioning accuracy high resolution images are needed, which requires higher time to be processed and, therefore, the frequency at which new data is available decreases –and the latency of such data increases–.

In these image-based positioning systems an agreement must been done between the accuracy of the positioning and the frequency at which the positioning data is obtained. Also the image acquisition is, in general, noisy, so the obtained results are also affected by this noise.

For the experimental setup, 800x600 pixels images are used, covering each camera an area of $\approx 3 \times 2$ meters, obtaining an accuracy $\approx 1\text{cm}$ at 10-15Hz using two ceil cameras simultaneously to cover a bigger area.

The visual analysis method used for this work is a variation of a color extraction algorithm with multiple elements to track. Each robot has a colour pattern located at the top of the robot as the one shown in figures B.5. These patterns are used to obtain the robot position and orientation inside the visual space of each camera. Inside this color pattern there is an area which contains the specific identifier of each robot, so the individual robots can be distinguished.

The basic flow of the visual location algorithm begins with the arrival of a new image and follows the general scheme represented at figure B.6. A sample of the unprocessed image can be seen in figure B.7

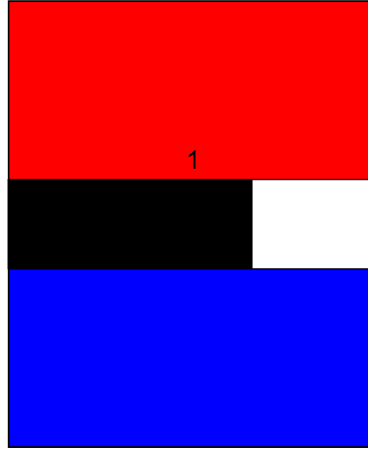


Figure B.5: Color Pattern on top of each robot, the black and white central strip identifies each robot

After a new image is obtained, the regions with the colors of the known pattern –red and blue– are extracted. The extraction is done by filtering individual pixels through a threshold algorithm working in the hsv color space, and building a connection map to set the individual pixels into regions.

Once the two color regions are extracted they are matched into pairs of potential patterns through the analysis of mutual distance, relative region size and shape.

Then, knowing the position and alignment of the potential patterns, the white and black strip located between the colour regions is used to obtain the identification number of the pattern.

The last step is to match the potential patterns with the estimated positions of the robots. The correlation of position and orientation between potential patterns and estimated positions of the robots is used to discard possible false pattern candidates or to accept pattern candidates with poor results in the identifier extraction. A depiction of the results obtained after the image processing can be seen at figure B.8

The conversion of local image space, at each camera, to real space is done through an initial calibration. The calibration process is done through a multi linear least square method where known coordinates (x, y) are matched against image space coordinates (u, v) through the parameters $u, v, u^2, u \cdot v, v^2, u^3, u^2 \cdot v, u \cdot v^2, v^3$ parameters. Using this calibration, with the used pixel density and the camera sensor noise, the obtained precision through the camera positioning is around $\pm 1cm$ in position and $\pm 10deg$ in orientation with $10-15Hz$ in frequency –when using two cameras on the same computer, each one provides images at $15Hz$ without processing–.

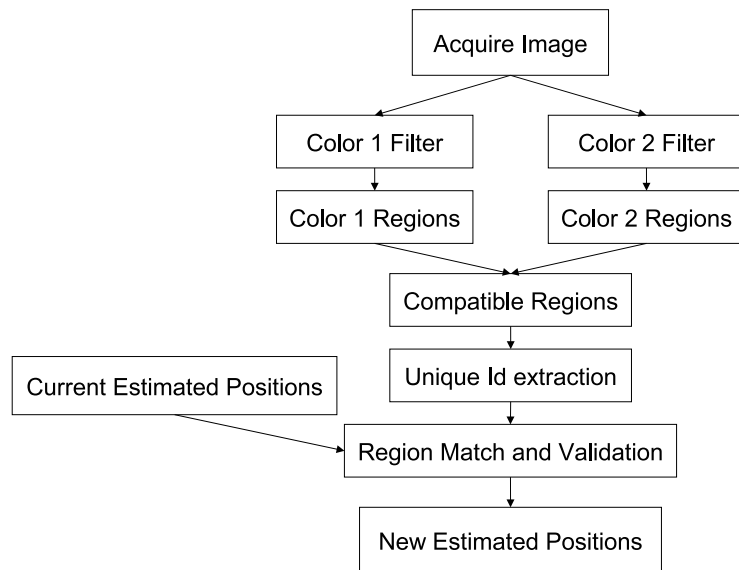


Figure B.6: Image Analysis process schematic diagram

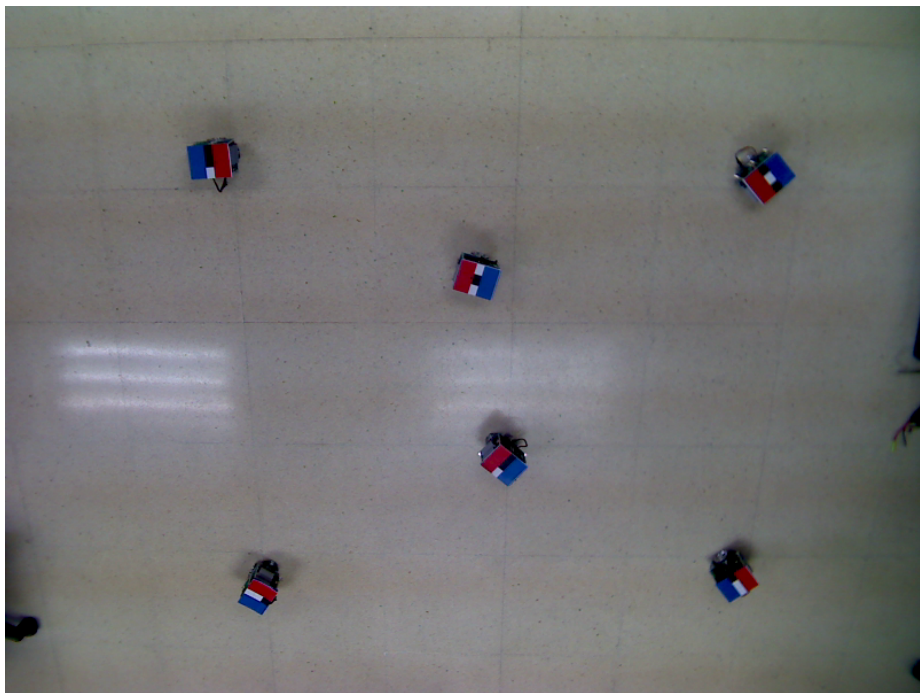


Figure B.7: Image captured by a ceil camera to be used used for robot location

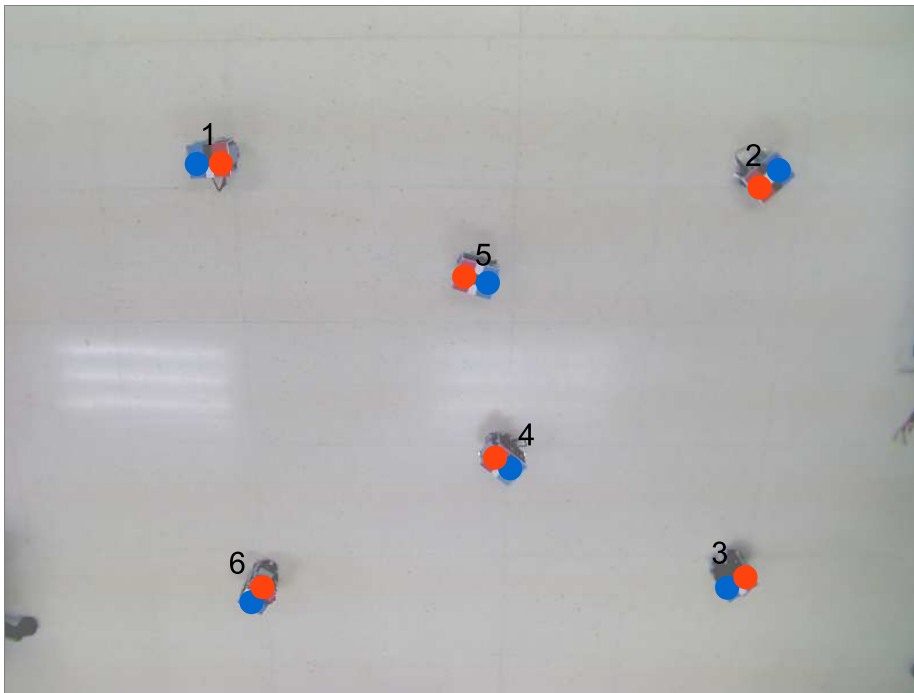


Figure B.8: Depiction of the results for the robot location through image analysis

B.3.3 Merging

The merge of the results obtained through the two employed methods, encoder motion tracking and absolute visual absolute positioning, provides a result which greatly reduces the problems of each independent method.

The main ideas behind the merging method used when these two independent methods are combined are two:

- The errors on the motion tracking obtained through encoders is small when the tracked interval is also small. So, when the absolute position of the robot is known –obtained from the camera–, for an small distance around this known position, the readings obtained from the encoders will be good and accurate.
- Knowing that for small distances the readings from the local encoders are accurate, the noise on the positioning resulting from the noise of the image acquisition process can be filtered out.

So, using the filtered position from the images as base location of the robot, the local encoders can be used to track the robot until the next filtered position of the image analysis is available. In this way, the error of the two methods are reduced and most of the advantages remains.

In order to put those two data sets together a simple Kalman propagation algorithm is used.

Some especial care has been taken due to the higher frequency on the position data obtained from the robot than the one obtained from the visual method, which also is subjected to lag. To solve this, the encoder readings from the robot and the images taken by the cameras are kept along with their acquisition time in the associated remote agent.

The robot position is propagated each time that a new encoders reading is available using the last accepted reference position.

When a new absolute positioning from the image processing is available, the encoder readings are propagated from the last accepted reference position and time, to the time at which the image –of the newly available absolute position– was acquired.

The two estimated absolute positions –one through the propagation of the encoder readings and one from the image analysis– are then merged using the Kalman algorithm. The resulting position –and error– obtained from the Kalman algorithm is the newly accepted reference position of the robot.

The new current position of the robot will be obtained by the propagation of the encoder readings from the newly accepted reference position and time.

This process is approximately depicted in equations at B.1. $Pos(t_{10})$ reflect how the robot position is obtained before a new image is available. At $t = 11$ the results of an image taken at $t = 6$ are available, $ImP(t_6)$. The availability of this new image results in the definition of a new reference position, $RefP(t_6)$ which corresponds to the accurate position of the robot at $t = 6$. Having the new reference position, the position obtained at $t = 11$, $Pos(t_{11})$ is obtained using this new reference position.

$$\begin{aligned}
Pos(t_{10}) &= RefP(t_0) + \sum_{0 \rightarrow 10} Enc(t_i) \\
RefP(t_6) &= Kalman \left(RefP(t_0) + \sum_{0 \rightarrow 6} Enc(t_i), ImP(t_6) \right) \\
Pos(t_{11}) &= RefP(t_6) + \sum_{6 \rightarrow 11} Enc(t_i)
\end{aligned} \tag{B.1}$$

Appendix C

Custom related works

The idea behind this Thesis, as many other ideas, have evolved along time and that evolution have been reflected in the different contributions done along the Thesis developing period.

Initially it was the aim of handling multiple robots, and the need of simulate them, so a simulation environment was developed and their main elements were shown at [Cifuentes06].

After that, the use of a independent field to keep several robots navigating together as a group was tested and shown at [Cifuentes08a].

Not later, that idea was generalized to handle the different elements of the scenario using different potential fields to keep the information isolated. This concept was tested and compared against the classical approach of using a single, unified, potential field, and the results were shown at [Cifuentes08b].

The next step on the application of the multiple potential field was its use in the formation navigation handling.

While the use of the expanded action space and the action blending techniques was present since [Cifuentes08a] it was not until [Cifuentes10] that its relevance was shown -in the previous texts the main emphasis was done in the use of multiple fields vs the use of a single field-.

In [Cifuentes10] the distinct actions and the blending process is finally shown to the public, however in a preliminary approach, and used to guide several robots in formation.

A more complete overview of the whole work was published in [Cifuentes12a] and [Cifuentes12b] were all the elements present at this Thesis were shown together and with a little more detail that the one allowed previously.

In the first one, [Cifuentes12a], the multiple action view and the action blending was introduced with proper detail, and then the scenario of a single robot and an obstacle explained using the proposed method.

The second article, [Cifuentes12b], explores the systematic use of the method for the navigation of a group of robots in formation. The same elements as the ones showed in this Thesis were present, however the emphasis was in the ability

of the method to handle many behaviours at once so the formation navigation scheme was done using tens of situations/actions sets.

All these previous texts holds the essential elements used along the proposed method, however the specifics of its application and the generalization of the ideas have been evolving along the process until it reaches this final Thesis document.

Bibliography

- [Arai02] Arai, T.; Pagello, E.; Parker, L.E. *Guest editorial advances in multi-robot systems* in IEEE Transactions on Robotics and Automation, vol 18, pp. 655 - 661 , Oct. 2002.
- [Arkin89] R. C. Arkin *Motor Schema-Based Mobile Robot Navigation* in the International Journal of Robotics Research, pp. 92-112, August 1989.
- [Arkin90] R.C. Arkin *Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation* in Robotics and Autonomous Systems, vol. 6, pp. 105-122, 1990.
- [Arkin98] R.C. Arkin *Behavior-Based Robotics* MIT Press, 1998.
- [Balch95] Balch, T.; Arkin, R.C. *Motor Schema-based Formation Control for Multiagent Robot Teams* in proc. of the First International Conference on Multi-Agent Systems, pp. 10-16, AAAI press, 1995.
- [Balch98] Balch, T.; Arkin, R.C. *Behavior-based Formation Control for Multi-robot Teams* in IEEE Transactions on Robotics and Automation, vol 14, pp. 926-939, Dec 1998.
- [Balch00] Balch, T.; Hybinette, M. *Social Potentials for Scalable Multi-Robot Formations* in proc. of the IEEE International Conference on Robotics and Automation, vol.1, pp. 73-80, San Francisco, 2000.
- [Borenstein89] J. Borenstein, Y. Koren *Real-time Obstacle Avoidance for Fast Mobile Robots* in IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept./Oct. pp. 1179-1187, 1989.
- [Borenstein90] J. Borenstein, Y. Koren *Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments* in Proc. of the IEEE International Conference on Robotics and Automation, vol. 1, pp.572 - 577, Cincinnati, 1990.
- [Brooks86] R. A. Brooks *A Robust Layered Control System For A Mobile Robot* in IEEE Journal of Robotics and Automation, vol. 2 n I, pp.14-23, March 1986.
- [Cifuentes06] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Modular software environment for general purpose autonomous agents simulation and control* in the 5th Mathmod, International Conference on Mathematical Modelling, Vienna, Austria, 2006.

- [Cifuentes08a] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Virtual Field Simulation Environment for Robot Groups Navigation Analysis* in the 7th IASTED International Conference, Innsbruck, Austria, 2008
- [Cifuentes08b] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Multiple Potentials Approach for Local Navigation of Mobile Robot Groups* in the International Symposium on Robotics, Seoul, Korea, 2008
- [Cifuentes10] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Robot Formation Control Based on a Multi-potential Approach* in the International Conference on Control, Automation and Systems, Gyeonggi-do, Korea, 2010
- [Cifuentes12a] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Robot Navigation Based on Discrimination of Artificial Fields: Application to Single Robots* in Advanced Robotics, Vol. 26 No. 6-7, April 2012.
- [Cifuentes12b] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Robot Navigation Based on Discrimination of Artificial Fields: Application to Robot Formations* in Advanced Robotics, Vol. 26 No. 6-7, April 2012.
- [Dudek96] Dudek, G.; Jenkin, M. R. M.; Milios, E.; Wilkes D. *A Taxonomy for Multi-Agent Robotics* in Autonomous Robots, vol 3, pp. 375-397, 1996.
- [Fredslund02] Fredslund, J.; Mataric M.J. *A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication* in IEEE Transactions on Robotics and Automation, vol 18, issue 5, pp. 837-846, Oct 2002.
- [Gat93] Gat, E. *Navigation templates: enhancements, extensions, and experiments* in Proc. of the IEEE International Conference on Robotics and Automation, vol. 1, pp. 541 - 547, Atlanta, 1993.
- [Ge04] Ge, S.S.; Fua, C.; Lim, K.W. *Multi-robot formations: queues and artificial potential trenches* in proc. of the IEEE International Conference on Robotics and Automation, vol. 4, pp.3345-3350, April 2004.
- [Gerkey02] Gerkey, B.P.; Mataric, M.J. *Sold!: Auction Methods for Multirobot Coordination* in IEEE Transactions on Robotics and Automation, vol. 8, Issue 5, pp 758-768, Oct 2002.
- [Guy10] Guy S.J.; Chhugani J.; Curtis S.; Dubey P.; Lin M.; Manocha D. *PLEdgestrians: A Least-Effort Approach to Crowd Simulation* in proc. of the Eurographics/ACM SIGGRAPH Symposium on Computer Animation, Madrid, Spain, 2010.
- [Hui10] N.B. Hui *Coordinated Motion Planning of Multiple Mobile Robots Using Potential Field Method* at the International Conference on Industrial Electronics, Control & Robotics (IECR), pp. 6-11, Orissa, 2010.
- [Kang11] Y.Kang; M.Lee; C.Kim; S.Yoon; C.Noh *A Study of Cluster Robots Line Formatted Navigation Using Potential Field Method* in the Proc. of the IEEE International Conference on Mechatronics and Automation, pp. 1723-1728, Beijing, 2011.

- [Khatib78] O. Khatib, J.F. LeMaitre. *Dynamic control of manipulators operating in a complex environments* in Proc. of the 3rd CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators, pp. 267-282, Udine, Italy, September 1978.
- [Khatib80] O. Khatib *Commande dynamique dans l'espace operationnel des robots manipulateurs en prsence d'obstacles*, PhD Thesis at Ecole Nationale Supriure de l'Aronautique et de l'Espace, Toulouse, France 1980.
- [Khatib85] O. Khatib *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots* in Proc. of IEEE International Conference on Robotics and Automation, pp. 500-505, St Louis, 1985
- [Khatib86] O. Khatib *Real time obstacle avoidance for manipulators and mobile robots* in Intl. J. Robotics Research, 5(1), pp. 90-99, 1986.
- [KhatibM96] M. Khatib *Sensor-based motion control for mobile robots* PhD Thesis at LAAS-CNRS, Toulouse, 1996.
- [Khosla98] P. Khosla, R. Volpe *Superquadric artificial potentials for obstacle avoidance and approach* in Proc. of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 1778-1784, Philadelphia, 1998.
- [Koren91] Y. Koren, J. Borenstein *Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation* in Proc. of the IEEE Intl. Conf. Robotics Automat., pp. 1398-1404, 1991.
- [Krogh84] B.H. Krogh *A Generalized Potential Field Approach to Obstacle Avoidance Control* at the Robotics International Robotics Research Conference, Bethlehem, PA, 1984.
- [Krogh86] B.H. Krogh, C. Thrope *Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles* in Proc. of the IEEE International Conference on Robotics and Automation, Vol. 3 pp. 1664-1669, 1986.
- [Lyons86] D. Lyons *Tagged potential fields: An approach to specification of complex manipulator configurations* at the IEEE International Conference on Robotics and Automation, 1984.
- [Mataric92b] M. Mataric *Behavior-Based Control: Main Properties and Implications* in the proc. of the IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems, pp. 46-54, 1992.
- [Michaud97] F. Michaud *Selecting behaviors using fuzzy logic* in proc. of the Sixth IEEE International Conference on Fuzzy Systems, vol.1, pp. 585-592, Barcelona, Spain, 1997.
- [Minguez04] J. Minguez, L. Montano *Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios* in IEEE Transactions on Robotics and Automation, vol 20, pp.45-49, Feb. 2004.
- [Moravec85] H. Moravec, A. Elfes *High resolution maps from wide angle sonar* in Proc. of the IEEE International Conference on Robotics and Automation, vol. 2 , pp. 116-121, 1985.

- [Naffin04] Naffin, D. J.; Sukhatme, G.S. *Negotiated Formations* in proc. of the International Conference of Intelligent Autonomous Systems -IAS-, pp.180-190, 2004.
- [Park03] M. G. Park, M. C. Lee *Artificial Potential Field Based Path Planning for Mobile Robots using a Virtual Obstacle Concept* in Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 735-740, Kobe, 2003.
- [Parker00] Parker, L.E. *Current State of the Art in Distributed Autonomous Mobile Robotics* in Distributed Autonomous Robotic Systems, pp. 3-12, Springer, 2000.
- [Prijanian99] P. Prijanian *Behavior Coordination Mechanisms - State-of-the-art* in Technical report IRIS-99-375, Institute of Robotics and Intelligent Systems, Univ. Southern California, 1999.
- [Quinlan93] S. Quinlan, O. Khatib *Elastic Bands: Connecting Path Planning and Control* in the Proc. of the IEEE International Conference on Robotics and Automation, vol.2 pp.802 - 807, Atlanta, 1993.
- [Reif99] J. H. Reif, H. Wang *Social potential Fields: A distribued behavioral control for autonomous robots* in Robotics and Autonomous Systems, Vol. 27, no.3, pp. 171-194, 1999.
- [Renaud76] M. Renaud *Contribution l'tude de la modelisation des systemes mcaniques articuls* PhD Thesis, Toulouse, 1976.
- [Reynolds87] Reynolds, C.W. *Flocks, Herds, and Schools: A Distributed Behavioral Model* in Computer Graphics, 21(4) -SIGGRAPH '87 Conference Proceedings- pp. 25-34, 1987.
- [Riecki96] J. Riecki, J. Roning *Reactive task execution by combining action maps* in proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '97, vol 1, pp 224-230, Grenoble , France, 1997.
- [Rosenblatt89] J. K. Rosenblatt, D. W. Payton *A fine-grained alternative to the subsumption architecture for mobile robot control* in proc. of the International Joint Conference on Neural Networks, vol 2, pp. 317-323, Washington DC, 1989.
- [Saffiotti95] A. Saffiotti, K. Konolige, E. H. Ruspini *A Multivalued Logic Approach to Integrating Planning and Control* in Artificial Intelligence, vol 76, pp. 481-526, 1995.
- [Saffiotti97] A. Saffiotti *The uses of fuzzy logic in autonomous robot navigation* in Soft Computing 1, pp. 180-197, Springer-Verlag, 1997.
- [Siemiatkowska94] B. Siemiatkowska *Cellular Neural Networks for Mobile Robot Navigation* at the Third IEEE International Workshop on Cellular Neural Networks and Their Applications, pp 18-21, 1994.
- [Slack91] Slack M.G. *Situationally Driven Local Navigation for Mobile Robots* Ph.D. Dissertation at Virginia Polytechnic Institute and State University, 1991.

- [Song02] P. Song, V. Kumar *A Potential Field Based Approach to Multi-Robot Manipulation* in the Proc. of the IEEE International Conference on Robotics and Automation, vol 2, pp. 1217-1222, 2002
- [Sugihara90] Sugihara, K.; Suzuki, I. *Distributed motion coordination of multiple mobile robots* in proc. of the 5th IEEE International Symposium on Intelligent Control, vol. 1, pp. 138-143, Philadelphia, USA, 1990.
- [Tan96] Tan, K.; Lewis, M.A. *Virtual structures for high-precision cooperative mobile robotic control* in proc. of the IEEE/RSJ International Conference Intelligent Robots and Systems, vol. 1, pp. 132-139, Nov 1996.
- [Thrope84] C. Thorpe, L. Matthies *Path Relaxation: Path Planning for a Mobile Robot* in the Proc. of the OCEANS, pp 576 - 581, Washington, 1984.
- [Tilden95] M.W. Tilden, B. Hasslacher *Living machines* in Robotics and Autonomous Systems, vol 15, pp. 143-169, July 1995.
- [Ulrich98] I. Ulrich, J. Borenstein *VFH+: reliable obstacle avoidance for fast mobile robots* in Proc. of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 1572-1577, Leuven , Belgium, 1998.
- [Ulrich00] I. Ulrich, J. Borenstein *VFH*: Local Obstacle Avoidance with Look-Ahead Verification* in the Proc. of the IEEE International Conference on Robotics and Automation, pp. 2505-2511, San Francisco, 2000.
- [Vail02] D.Vail, M. Veloso *Multi-Robot Dynamic Role Assignment and Coordination Through Shared Potential Fields*, on Multi-Robot Systems, pub. Kluwer, 2003.
- [Walter53] W. G. Walter, *The Living Brain*, Norton, New York, 1953.
- [Yun96] Yun, X.; Alptekin, G.; Albayrak O. *Line and circle formation of distributed physical mobile robots* in Journal of Robotic Systems 14(2), pp. 63-76, 1997.

List of Figures

2.1	The electric dog and a test showing how it follows the light . . .	12
2.2	Elise, Walter's second tortoise. One of the first robots with multiple behaviours	12
2.3	Robot arm movement result exposed in the work of Kathib . . .	14
2.4	Potential field on a scenario with several object, its representation is like covering everything with a soft sheet	15
2.5	Polar map built out of the Certainty Grid representation and flat result from the work of Borenstein	16
2.6	Deliverative Vs Reactive Approaches, from L. E. Parker courses .	18
2.7	Classic approach –using robot dynamic equations– and Behavioural approach –evaluating objectives independently–.	19
2.8	Uni-Valued and Multi-Valued output from an <i>Avoid Collision</i> behaviour	21
2.9	Basic Formation Structures: Virtual Leader, Neighbours and Virtual Structure. Each robot is associated with each of the desired locations	24
2.10	Anonymous Formation Structures: Geometry based, Function Minimizing and Social Fields. Each robot have the same role and its position depends on the whole group situation	25
3.1	Single Robot with obstacle ahead. Classic approach with single valued actions applied directly on the actuators	32
3.2	Single Robot with obstacle ahead. Proposed approach with multi-valued actions with an increased actions space	33
3.3	Preview of the single robot scenario.	35
3.4	Preview of the multiple of robots scenario.	35
3.5	Preview of the unstructured group of robots scenario.	35
3.6	Sample of the process to reach a formation structure -wedge-using 11 robots	36
4.1	Goal and single obstacle scenario with its resulting field	41
4.2	Contour and side cut of the field value for a simple scenario with a goal and an obstacle	41
4.3	Goal and Obstacle fields surface keep independent	42
4.4	Goal and Obstacle fields contours keep independent	42
4.5	Descriptive Element and the emphasizing result of the power operation	45
4.6	<i>Obstacle Near</i> : $d_{Near} = x_{099} = 20cm$, $d_{Far} = x_{001} = 50cm$. . .	46

4.7	Linear and a Non-Linear normalizations of an angular measure .	46
4.8	Linear and a Non-Linear normalizations of an angular measure, polar representation	47
4.9	Evaluation of $S_{CollisionDanger}$ for different relative obstacle posi- tions	48
4.10	Equivalent obstacle sensed by the robot when several real obsta- cles are present in the scenario	48
4.11	Classic Gaussian Curve and Extended Gaussian Curve used to define the actions	51
4.12	Polar representation of the classic Gaussian Curve and Extended Gaussian Curve used to define the actions	51
4.13	<i>Elected</i> action map built from three <i>Elect</i> actions. Cartesian and Polar representations	54
4.14	Example of the union of three contributions. Cartesian and Polar representations	54
4.15	Example of <i>Suitability</i> map along with its source <i>Elected</i> and <i>Forbidden</i> maps. Cartesian and Polar representations	55
4.16	Example of <i>Safety</i> map along with its source <i>Forbidden</i> map. Cartesian and Polar representations	57
4.17	Example of <i>Turning</i> map along with its source <i>Suitability</i> and <i>Safety</i> maps. Cartesian and Polar representations	57
5.1	Single simulated robot scenario and visualization of the employed Fields	64
5.2	Robot path along the classic local minima trap scenario using the proposed method. Suitability map represented around the robot at each capture.	65
5.3	Evolution of situation indicators $S_{1,0}$ (solid) and $S_{1,1}$ (dash) com- puted by the robot along the simulation time	66
5.4	Descriptive relative positions between the sensing robot and the equivalent robot in polar representation	68
5.5	Two robots in the proximity of the sensing one in two configura- tions showing the perceived robot in a good and bad representa- tion of the real ones	69
5.6	Multiple independent robots, capture 1. Initial positions of the robots, goals and obstacle for the multiple robots example. . . .	73
5.7	Multiple independent robots, capture 2.	74
5.8	Multiple independent robots, capture 3.	75
5.9	Multiple independent robots, capture 4.	76
5.10	Multiple independent robots, capture 5.	77
5.11	Multiple independent robots, capture 6.	77
5.12	Robot 3 Situations.	78
5.13	Robot 3 Speed.	78
5.14	Situation $S_{3,21}$ depiction, when team and goal directions are aligned, the robot is at the back of the group	83
5.15	Situation $S_{3,22}$ depiction, when team and goal directions are op- posed, the robot is ahead of the group	84
5.16	Simulation captures along the path for an unstructured robot group	85
5.17	$Team_{Distant}$ descriptive element for the detached robot along the group simulation	86

5.18	Group related situation indicators for the detached robot along the group simulation	87
5.19	Formations defined through traces vs. formations defined by dots	89
5.20	Formation field built from a formation trace	90
5.21	Line formation field	90
5.22	Column formation field	91
5.23	Wedge formation field	91
5.24	Circle formation field	92
5.25	Normalized descriptor for formation distances	96
5.26	Relative position of the robot within the formation	97
5.27	Descriptive elements for sensed robot position relative to leading direction in polar representation	98
5.28	5 Robots arranging a line formation	104
5.29	7 Robots arranging a line formation	105
5.30	11 Robots arranging a line formation	107
5.31	20 Robots arranging a line formation	108
5.32	5 Robots arranging a column formation	110
5.33	7 Robots arranging a column formation	111
5.34	11 Robots arranging a column formation	112
5.35	20 Robots arranging a column formation	113
5.36	5 Robots arranging a circle formation	115
5.37	7 Robots arranging a circle formation	116
5.38	11 Robots arranging a circle formation	117
5.39	20 Robots arranging a circle formation	118
5.40	5 Robots arranging a wedge formation	120
5.41	7 Robots arranging a wedge formation	121
5.42	11 Robots arranging a wedge formation	122
5.43	20 Robots arranging a wedge formation	123
5.44	Line formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations	125
5.45	Column formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations	126
5.46	Circle formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations	127
5.47	Wedge formation, Mean robot distance to formation along simulation time \pm Std of 101 simulations	128
5.48	Coefficients from the fitting of the distance to formation along simulation time to a exponential function for different number of robots in the formation	130
5.49	Coefficients from the fitting of the standard deviation on distance to formation along simulation time to a exponential function for different number of robots in the formation	132
5.50	Line formation, Formation arrangement time histograms for 101 simulations	134
5.51	Column formation, Formation arrangement time histograms for 101 simulations	135
5.52	Circle formation, Formation arrangement time histograms for 101 simulations for 101 simulations	136
5.53	Wedge formation, Formation arrangement time histograms for 101 simulations	137

5.54	Formation Arrangement Times, Mean \pm Std	139
5.55	Line formation, Mean distance to formation along simulation time \pm Std, Noisy simulation	140
5.56	Column formation, Mean distance to formation along simulation time \pm Std, Noisy simulation	141
5.57	Circle formation, Mean distance to formation along simulation time \pm Std, Noisy simulation	142
5.58	Wedge formation, Mean distance to formation along simulation time \pm Std, Noisy simulation	143
5.59	Coefficients from the fitting of the distance to formation along simulation time to a exponential function, Noisy simulation	144
5.60	Coefficients from the fitting of the standard deviation on the distance to formation along simulation time to a exponential function, Noisy simulation	145
5.61	Coefficients from the fitting of the distance to formation along simulation time to a exponential function, Noisy Vs Noiseless, simulation	146
5.62	Coefficients from the fitting of the standard deviation on the distance to formation along simulation time to a exponential function, Noisy Vs Noiseless, simulation	147
5.63	Line formation, Formation arrangement time histograms, Noisy simulation	148
5.64	Column formation, Formation arrangement time histograms, Noisy simulation	149
5.65	Circle formation, Formation arrangement time histograms, Noisy simulation	150
5.66	Wedge formation, Formation arrangement time histograms, Noisy simulation	151
5.67	Formation Arrangement Times, Mean \pm Std, Noisy simulation . .	153
5.68	Column formation, slow turn (0.01rad/s) for 5 robots	155
5.69	Column formation, medium speed turn (0.05rad/s) for 5 robots .	156
5.70	Column formation, sharp turn (1rad/s) for 5 robots	157
5.71	Circle formation, sharp turn (1rad/s) for 11 robots	158
5.72	Distance to formation for various turn ratios, 5 robots in line formation.	160
5.73	Distance to formation for various turn ratios, 5 robots in column formation.	161
5.74	Distance to formation for various turn ratios, 5 robots in circle formation.	162
5.75	Distance to formation for various turn ratios, 5 robots in wedge formation.	163
5.76	Formation Transition, Line to Wedge formation with 5 robots . .	165
5.77	Formation Transition, Wedge to Column formation with 5 robots	166
5.78	Formation Transition, Column to Line formation with 5 robots .	167
5.79	Formation Transition, Line to Circle formation with 5 robots . .	168
5.80	Line formation vs. Column obstacle with 9 robots	170
5.81	Column formation vs. Wall obstacle with 9 robots	171
5.82	Circular formation vs. Column obstacle with 9 robots	172
5.83	Diamond virtual structure with 9 robots	174
5.84	Smiling face virtual structure with 9 robots	175

6.1	Line formation with 3 robots	178
6.2	Column formation with 3 robots	179
6.3	Circle formation with 3 robots	180
6.4	Wedge formation with 3 robots	181
6.5	Line formation with 4 robots	182
6.6	Column formation with 4 robots	183
6.7	Circle formation with 4 robots	184
6.8	Wedge formation with 4 robots	185
6.9	Line formation with 5 robots	186
6.10	Column formation with 5 robots	187
6.11	Circle formation with 5 robots	188
6.12	Wedge formation with 5 robots	189
6.13	Line formation with 6 robots	190
6.14	Column formation with 6 robots	191
6.15	Circle formation with 6 robots	192
6.16	Wedge formation with 6 robots	193
6.17	Experimental mean distance to formation and its standard deviation for 4 robots	195
6.18	Experimental Arrangement Times for four robots	196
6.19	Experimental mean distance to formation fitting coefficients against coefficients obtained at the noisy simulations	198
6.20	Experimental mean distance to formation deviation fitting coefficients against coefficients obtained at the noisy simulations	199
A.1	Simulation Loop for sequential <i>devices</i>	206
A.2	Simulation environment visual interface	209
B.1	Experimental Robot	212
B.2	UsbLab control board	213
B.3	Radio board	214
B.4	Motor Driver board	215
B.5	Color Pattern on top of each robot, the black and white central strip identifies each robot	218
B.6	Image Analysis process schematic diagram	219
B.7	Image captured by a ceil camera to be used used for robot location	219
B.8	Depiction of the results for the robot location through image analysis	220

List of Tables

5.1	Situation $S_{1.0}$ related actions	62
5.2	Situation $S_{1.1}$ related actions	63
5.3	Descriptive elements related with the sensed robot field orientation	68
5.4	Actions related with situation $S_{2.1}$	71
5.5	Actions related with situation $S_{2.2}$	72
5.6	Actions related with situation $S_{2.1}$ for Robot Groups	81
5.7	Actions related with situation $S_{3.1}$ for groups	82
5.8	Actions related with situation $S_{3.21}$ for groups	83
5.9	Actions related with situation $S_{3.22}$ for groups	84
5.10	Actions related with situation $S_{3.23}$ for groups	84
5.11	Approaching Actions $S_{3.1}$ for formations	99
5.12	Special Symmetry avoidance in $S_{3.1}$ for formations	100
5.13	Formation keeping actions related with situation $S_{3.2}$ for formations	101
5.14	Formation insertion actions related with situation $S_{3.2}$ for formations	101
5.15	Formation Arrangement Times for the different Formation Shapes and number of robots used in the simulation, without noise . . .	139
5.16	Formation Arrangement Times for the different Formation Shapes and number of robots used in the simulation, Noisy simulation .	152



UNIVERSIDAD COMPLUTENSE
MADRID

FACULTAD DE CIENCIAS FÍSICAS

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y
AUTOMÁTICA

TESIS DOCTORAL
–RESUMEN–

**Integración de Comportamientos para la
Navegación Coordinada de Múltiples Robots
mediante Potenciales Virtuales**

**Behaviour Blending for
Multiple Robot Coordinated Navigation
through Virtual Potential Fields**

por Santiago Cifuentes Costa

Directores:
Profesor Dr. Jose Maria Giron Sierra
Profesor Dr. Juan Francisco Jimenez

Índice general

1. Introducción	5
2. Resumen General	9
2.1. Inicio y desarrollo del tema	9
2.2. Propuesta principal	11
2.2.1. Combinación de acciones	11
2.2.2. Descripción de las acciones	13
2.2.3. Modelado del mundo	13
2.3. Casos de estudio	15
2.4. Resultados	19
2.5. Publicaciones	19
3. Detalles del Método	21
3.1. Robots Móviles	21
3.2. Elementos del Algoritmo	21
3.2.1. Campos de Potencial	21
3.2.2. Percepción y manejo de datos	23
3.3. Proceso de decisión	24
3.3.1. Descripción de las Acciones	24
3.3.2. Acciones empleadas	25
3.3.3. Superposición de acciones	25
3.3.4. Procesado de acciones y composición final	26
4. Casos de Estudio	31
4.1. Robot y Obstáculo	31
4.2. Navegación en formación	34
4.2.1. Campos	34
4.2.2. Elementos descriptivos, Situaciones y Acciones	34
5. Análisis de resultados	39
5.1. Simulación	39
5.2. Robots Reales	43
6. Conclusiones	45
6.1. Aplicaciones y trabajo futuro	46
Bibliography	47

Capítulo 1

Introducción

Los robots son máquinas asombrosas, capaces prácticamente de todo, desde hacer labores domésticas, ayudar a nuestros mayores, realizar el trabajo duro por nosotros e incluso salvar el mundo! –o a veces conquistarlo– ... al menos, esa es la imagen que tenemos de ellos a través de las películas, los libros de ciencia ficción y cualquier historia semi-futurística.

Lo cierto es que los robots son máquinas maravillosas, capaces de un montón de cosas, pero aun están en desarrollo y necesitan bastante trabajo antes de ser capaces de tan magníficas proezas como les atribuimos.

En las últimas décadas hemos experimentado un gran aumento en la ayuda que nos proporcionan las máquinas, especialmente las electrónicas, tanto en la vida diaria como en la industria. Radio, televisión, neveras, aspiradores, ordenadores, reproductores de música, teléfonos móviles... nos hemos acostumbrado a la tecnología.

A cada paso de esta adaptación, esperamos que la tecnología dependa cada vez menos de nuestro conocimiento sobre ella y que sea capaz de adaptarse a nuestras necesidades con un mínimo conocimiento por nuestra parte.

Cada vez esperamos más independencia y autonomía de las máquinas que usamos. En parte porque especificar todos los pequeños detalles y resolver las pequeñas inconsistencias sobre lo que queremos que hagan las máquinas es muy pesado, necesita aprendizaje sobre elementos de la máquina que realmente no nos interesan y hace que su uso no sea agradable. No ayuda, complica.

En parte es debido a la forma en la que elaboramos nuestras ideas, y por tanto, la forma en que las expresamos.

Cuando pensamos acerca de cómo resolver un problema suficientemente complejo, no es habitual que inmediatamente consideremos todos los detalles, las interacciones y los diferentes aspectos implicados en el problema.

La forma más habitual es tomar el problema por partes, solucionar individualmente cada parte y luego refinar el conjunto de soluciones buscando las incompatibilidades entre ellas. El conocimiento experto de un problema es nece-

sario precisamente para solucionar dichas incompatibilidades y sobre todo para asegurarse de que se han considerado todos los aspectos implicados.

Por tanto, lo que esperamos son máquinas con conocimiento experto sobre el tema en el que trabajan. Máquinas que sean capaces de trabajar a partir de nuestras ideas iniciales, independientes y sin refinar. Esperamos que sea la propia máquina la que resuelva los pequeños conflictos y rellene los huecos que dejamos.

La idealización de este tipo de máquina experta son los robots. Probablemente llegaremos a un punto donde los robots sean, comparados con nuestros actuales ordenadores, como son los SmartPhones comparados con la radio de Marconi. Pero para ésto, aún hemos de resolver algunos problemas, siendo uno de ellos la capacidad para tomar un conjunto de ideas iniciales –incompletas y con sus pequeñas contradicciones– y ponerlas en conjunto para hacer la tarea que esperamos de ellos –que no la que les hemos dicho–

El principal punto débil de la robótica hoy en día está en el lado de la lógica del robot. La forma en la que la información de los sensores debe analizarse, la forma de coordinar los diferentes actuadores y, principalmente, la forma en la que dotar a los robots de la inteligencia que se les asocia para que sean robots y no simplemente máquinas automáticas.

Un elemento que falta en esta supuesta inteligencia es, en ciertos aspectos, el poder poner a trabajar juntos todos los elementos del robot, y no que el robot sea un conjunto de elementos independientes. El problema no está normalmente en la falta de capacidad de proceso del robot, sino en el método –o más bien la falta de un buen método– para poder describirle al robot los comportamientos que deseamos que muestre.

Normalmente lo que se encuentra en la mayoría de los robots es que resuelven cada problema de una forma específica, y cuando el robot debe enfrentarse a varios problemas a la vez, las diferentes formas de resolver cada problema no son compatibles entre sí, o requieren el tener que replantearlas completamente para poder atacar el problema general.

Propuesta Una solución válida para todos los posibles aspectos de la robótica no es fácilmente alcanzable debido a la amplitud de aplicaciones que ésta tiene. Sin embargo, la exploración de posibles soluciones generales en aplicaciones específicas nos da la oportunidad de encontrar un principio que pueda ser expandido y aplicado a otras áreas de trabajo.

A lo largo de esta Tesis, se propone una forma de expresar las diferentes condiciones del entorno en el que se encuentra el robot, y de enunciar las diferentes acciones para que éstas, expresadas independientemente, puedan ser coordinadas de modo que el robot lleve a cabo su tarea.

El campo de aplicación del trabajo realizado es la navegación, un elemento fundamental presente en la mayoría de robots.

Aún cuando la propuesta principal de la Tesis es una forma de coordinar los diferentes elementos que debe tener cuenta un robot para navegar siguiendo un objetivo definido, las aplicaciones del método propuesto mostradas a lo largo

de la Tesis van desde la navegación de un único robot hasta la navegación coordinada de múltiples robots moviéndose en conjunto.

Estructura de la Tesis En el primer capítulo, se intenta ofrecer una visión general sobre la problemática del tema a tratar así como una idea de porqué es interesante el estudio del tema planteado. Ésto se acompaña de una breve descripción de cómo se desarrolla el texto de la Tesis.

A este capítulo introductorio le sigue una revisión de los principales autores y temas que tienen relevancia para esta Tesis. Esta revisión del arte previo está acompañada de breves notas sobre cómo los diferentes aspectos comentados afectan, influyen, o son tratados en la propuesta hecha por esta Tesis.

A continuación, en el tercer capítulo, se puede encontrar una serie de reflexiones sobre la evolución del arte previo y en qué elementos se puede considerar necesario profundizar. Además, con los principales elementos ya introducidos por el arte previo, se bosqueja de forma simple y aún sin formalismos, la metodología propuesta, los elementos necesarios para su aplicación, las pruebas realizadas y los resultados obtenidos. De este modo se da una visión de conjunto del trabajo propuesto y realizado.

En el cuarto capítulo es donde se realiza la propuesta formal de la Tesis. Con una aproximación que intenta ser generalista, se exponen los diferentes elementos que integran la metodología propuesta y a continuación se expone – de una forma general pero centrada en la navegación – un método para su puesta en práctica.

A lo largo del quinto capítulo, se presentan una serie de escenarios donde la metodología propuesta y su forma de aplicación son empleados.

La complejidad de los escenarios presentados va desde la navegación de un único robot evitando un obstáculo hasta la navegación en formación de un grupo de robots capaces de distribuirse en cualquier forma.

El planteamiento de los escenarios donde el método es aplicado –con una dificultad incremental– permite mostrar de forma aislada los diferentes elementos que están siendo tomados en consideración para la navegación del robot, y cómo se van aplicando estos nuevos elementos bajo el punto de vista del método propuesto. El incremento en complejidad de los escenarios permite también mostrar como este método permite añadir los nuevos elementos en consideración sin necesidad de rediseñar completamente la estructura lógica del robot.

Mientras que los escenarios en los que se aplica el método a lo largo del quinto capítulo son todos escenarios simulados, en el sexto capítulo se evalúa el método propuesto empleando robots reales.

Para la puesta en práctica sobre robots reales se emplea de nuevo el escenario más complejo evaluado en simulación: la navegación en formación de un grupo de robots.

La comparación de resultados en este escenario entre los ensayos en simulación y los ensayos con robots reales permite tanto demostrar la viabilidad

del método para su aplicación en sistemas reales como validar los resultados obtenidos mediante simulación.

En el séptimo capítulo se aportan algunas conclusiones extraídas a la vista de los resultados obtenidos en los capítulos anteriores. Aquí también se desarrollan brevemente algunas ideas acerca de las posibles aplicaciones del método y la metodología estudiados, y cómo puede emplearse para expandir esta u otras líneas de investigación.

Finalmente, en los apéndices, se muestran algunos detalles de las herramientas empleadas para la simulación, así como sobre los robots desarrollados para las pruebas reales y como éstas han sido realizadas.

Contribución El trabajo realizado se basa en muchos desarrollos anteriores. Toma ideas de distintos autores, adaptando las contribuciones más relevantes y tomando aquellas que mejor encajan con el objetivo propuesto.

La mayoría de los aspectos individuales de la Tesis no pueden considerarse originales, lo que puede considerarse original es la forma en la que han sido combinados en una metodología de aplicación y un método de implementación que, siguiendo unos mismos principios, es capaz de resolver de forma sencilla una gran variedad de problemas.

Además de esto, los casos de estudio empleados para probar la viabilidad del método propuesto no son todos casos con una solución establecida. De esta forma, la metodología y el método propuesto a lo largo de esta Tesis aportan una sólida solución a estos casos, con un nivel de generalidad suficiente para esperar unos resultados similares en aquellos casos con condiciones distintas a las planteadas.

Capítulo 2

Resumen General

A lo largo de la revisión del arte previo se muestra la cantidad de contribuciones que existen sobre los diferentes aspectos en la robótica, sin siquiera salir del tema de la navegación. Normalmente los trabajos se centran sobre un aspecto determinado o están enfocados en una tarea concreta. Las soluciones alcanzadas por estos trabajos muestran un cierto sesgo en la dirección de su objetivo, haciendo muchas veces difícil el distinguir el límite entre la parte dedicada puramente a la navegación y la parte asociable a la tarea en sí.

Si bien en general los trabajos son de una enorme calidad y muestran magníficos resultados, en ocasiones se echa en falta una cierta generalización del método empleado o propuesto. ¿Qué elementos de la tarea son los que influyen directamente en la navegación? ¿Qué dependencia existe entre el objetivo y la forma en que se lleva a cabo la navegación? ¿Cuál es la información mínima necesaria –no la óptima– para poder resolver el objetivo impuesto? Y sobre todo ¿Existe alguna forma en que se puedan tratar los casos resueltos de forma sistemática con la metodología propuesta? ¿Cómo se pueden identificar los elementos mínimos necesarios de un caso cualquiera para aplicar sistemáticamente el método propuesto?

A lo largo de este trabajo, se trata de identificar cada uno de los elementos integrantes del método propuesto, cómo aplicarlos de forma sistemática y su influencia para con los casos estudiados insertándolos de forma paulatina a lo largo de los diferentes objetivos.

2.1. Inicio y desarrollo del tema

Como en muchos otros casos, esta Tesis comenzó con el estudio de un tema ligeramente diferente: Buscar una forma de manejar uno o muchos robots, con diferentes relaciones estructurales, como una única entidad.

A lo largo de esa investigación inicial surgió la necesidad de identificar cuáles eran los elementos comunes de los diferentes casos que se estaban estudiando, qué elementos eran propios de cada caso y los efectos que tenían todos estos elementos sobre el proceso de navegación.

En ese momento el estudio inicial se centró en analizar cómo estructurar y hacer uso de los distintos elementos identificados, para establecer y describir de forma sistemática las diferentes relaciones entre los robots de un grupo y su navegación.

En ese punto se realizó un análisis exhaustivo para separar aquellos elementos que afectaban a la navegación en sí de los elementos que, afectando a la forma en que se llevaba a cabo la navegación, se correspondían con el objetivo concreto del caso en estudio.

Tras esta clasificación, el objetivo inicial de establecer una forma para manejar varios robots como una única entidad pasó a un segundo plano. Se pudo ver que las relaciones de estructura del grupo de robots se podían considerar elementos particulares del caso de estudio y que no estaban intrínsecamente relacionados con el proceso de navegación en sí.

Tras descomponer cómo los diferentes autores especificaban la navegación en sus robots, se pudo alcanzar una idea general sobre los elementos necesarios para la navegación y los elementos, específicos de cada caso, que hacían uso de la navegación.

La mayoría de los trabajos previos estudiados expresaban su métodos en términos del enfoque mediante *comportamientos* usado por Arkin y Brooks debido a que la descomposición de la tarea principal en múltiples subtareas –conceptualmente independientes– hace más fácil el expresar la complejidad total de la tarea principal.

Sin embargo, en el momento de la implementación del método propuesto, las subtareas no son siempre empleadas de forma independiente. Se observa que las subtareas empiezan a depender unas de otras cuando se lleva a cabo su aplicación sobre los actuadores del robot, debido a que aunque son conceptualmente independientes, en su aplicación es necesario establecer reglas de precedencia o de balanceado entre ellas.

Por tanto, cuando muchos de los métodos propuestos quieren aplicarse a otros casos distintos, la implementación del método necesita ser reescrita completamente.

La dificultad reside en general en la forma en que son expresados los términos de un caso específico, mezclando los principios de la navegación en sí con los objetivos concretos del caso estudiado. Ésto da como resultado que las diferentes consideraciones del caso específico se mezclen con las consideraciones de la navegación en sí, de modo que no es posible describir de forma independiente las consideraciones específicas del caso estudiado porque deben estar sometidas a las necesidades de la navegación.

El enfoque clásico de expresar las necesidades del caso estudiado en la forma directa de su aplicación sobre el espacio de actuadores, no permite el tipo de resultados buscados –poder describir las cosas de forma aislada– porque es precisamente este enfoque el que lleva a tener que enredar los distintos elementos.

2.2. Propuesta principal

Es objetivo de esta Tesis el proponer y mostrar un método que permita mantener independientes los diferentes elementos que deben tenerse en cuenta al describir el objetivo de un robot o grupo de ellos. Mantener independientes los aspectos concretos del caso estudiado, así como independientes de los elementos propios de la navegación.

El método propuesto tiene dos partes bien diferenciadas : Cómo expresar las acciones necesarias para llevar a cabo la tarea y cómo aplicar estas acciones sobre el espacio de actuadores.

2.2.1. Combinación de acciones

La principal característica del método propuesto es que el espacio de acciones –donde se aplican los resultados del análisis de las diferentes subtareas– y el espacio de actuadores –las capacidades físicas del robot– no son el mismo: el espacio de acciones tiene mayor dimensión que el espacio de actuadores.

El objetivo de este incremento en la dimensión del espacio de acciones es el aumentar la información contenida en el sistema, proporcionando así una forma de evitar conflictos que tendrían lugar si la acciones se hubiesen expresado directamente en el espacio de actuadores y, de esta forma, facilitar la manera de expresar el comportamiento del robot.

En este trabajo, la forma de evitar las actuaciones que darían lugar a conflicto es mantener en el sistema una cierta información sobre los motivos de las acciones propuestas y, mediante el uso de esta información, facilitar el proceso de combinación de las diferentes acciones para su aplicación sobre los actuadores del robot.

Una forma sencilla de ilustrar la idea propuesta es seguir el proceso que tiene lugar en un robot que se dirige hacia un objetivo y encuentra un obstáculo en su camino:

- Cuando se expresan las acciones del robot directamente en el espacio de actuadores aparece un conflicto entre las dos subtareas del robot como queda ilustrado en la figura 2.1:

La subtarea encargada de guiar el robot hacia el objetivo le indica al robot que se mueva hacia adelante, porque el objetivo está adelante. Al mismo tiempo la tarea encargada de evitar colisiones le indica al robot que se dirija hacia atrás –o en cualquier dirección menos hacia adelante– dado que el obstáculo está adelante.

Las dos actuaciones indicadas por las subtareas están en conflicto, pero ambas son perfectamente válidas. Dado que no se mantiene ninguna información sobre el motivo de cada una de ellas, a la hora de aplicar la acción sobre los actuadores del robot no se puede establecer cuál de las dos acciones es la que se debe aplicar.

–La solución inmediata que se nos puede ocurrir ante este problema sería el establecer la prioridad de la subtarea de evitar obstáculos sobre la subtarea de ir hacia el objetivo, sin embargo esto lleva directamente hacia

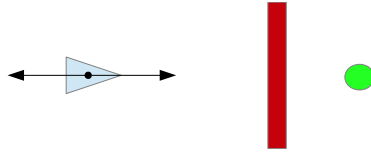


Figura 2.1: Robot con obstáculo al frente. Ejemplo de la descripción clásica mediante valores puntuales directamente en el espacio de actuadores.

el enredo entre subtareas que se pretende evitar. La solución de establecer una estructura jerárquica entre subtareas es el método propuesto por Brooks con sus *subsumptions*, pero está probado que el uso de estructuras jerárquicas sólo funciona para casos sencillos. Cuando la complejidad del conjunto de subtareas aumenta, es necesario establecer un mecanismo de asignación dinámica de prioridades, pero ese mecanismo es específico del conjunto de subtareas sobre el que actúa.

- Mediante el principio defendido por esta Tesis las dimensiones del espacio de acciones son mayores que las dimensiones del espacio de actuación. De esta forma podemos aumentar la actuación *mover* en dos acciones distintas pero ambas enfocadas a mover el robot: *intenta-mover* y *prohibido-mover*.

Empleando las dos acciones definidas, es inmediato como aplicarlas a la situación estudiada. La subtarea encargada de guiar el robot hacia el objetivo aplicará una acción del tipo *intenta-mover* hacia adelante, mientras que la subtarea encargada de evitar las colisiones aplicará una acción del tipo *prohibido-mover* hacia adelante.

En el momento en que estas acciones son puestas en común para decidir qué actuación debe aplicarse sobre los actuadores del robot –cómo debe moverse– ya no existe conflicto –*prohibido-mover* tiene prioridad sobre *intenta-mover*– pues la naturaleza de las acciones aporta una motivación sobre cada acción.

– En cierta forma el principio aplicado puede considerarse como una aplicación de una estructura de prioridades sobre las diferentes dimensiones del espacio de acciones. La diferencia con el establecimiento de jerarquías entre subtareas es que la estructura de prioridades ya no está relacionada con las subtareas, y por tanto puede seguir empleándose en casos con diferentes subtareas que puedan expresarse con las mismas acciones. –

El incremento en la dimensión del espacio de acción sobre el espacio de actuación viene determinado por el problema y las necesidades de las subtareas. Las relaciones entre las distintas dimensiones del espacio de acción para su combinación en el espacio de actuación se definen simplemente por conveniencia en la forma de expresar las acciones de las subtareas tratadas.

En esta Tesis, al ser aplicada sobre problemas de navegación de robots, es suficiente con el incremento en dimensión de la actuación *mover* en las acciones *intenta-mover* y *prohibido-mover* –donde *prohibido-mover* modula a *intenta-mover*–. Este aumento es suficiente para incrementar notablemente la facilidad con la que expresar los diferentes elementos necesarios para guiar al robot.

De esta forma, son tan sólo necesarias ocho acciones, independientes entre sí, para formar y mantener a los robots dentro de una formación cualquiera. Los elementos necesarios para evitar obstáculos y la colisión con otros robots se expresan, también de forma independiente, con tan solo otras cinco acciones.

2.2.2. Descripción de las acciones

Un segundo aspecto importante del método propuesto es la forma en la que se expresan las acciones que definen el comportamiento del robot.

El uso habitual de expresar las acciones/actuaciones empleando valores puntuales limita notablemente el conjunto de opciones válidas para ser elegidas en el proceso de decisión –cómo aplicar las distintas acciones definidas por las distintas subtareas sobre los actuadores finales del robot–.

Sin embargo, cuando las acciones son expresadas mediante colecciones de valores –o regiones– el conjunto de opciones válidas sobre las que elegir durante el proceso de decisión se amplía, sin por eso añadir mucha complejidad a la forma de expresar las acciones.

Tomando de nuevo el ejemplo del robot que se encuentra con un obstáculo en su camino hacia el objetivo, se puede ilustrar la idea previa:

- Cuando las acciones *intenta-mover* adelante y *prohibido-mover* adelante se expresan definiendo *adelante* como un único valor –dirección 0 grados–, al tomar la decisión de cómo mover el robot y aplicar la precedencia de *prohibido-mover* sobre *intenta-mover*, el robot se quedará quieto, pues no tiene ninguna dirección viable en la que moverse.
- Las dos mismas acciones pueden definirse mediante un rango de valores en lugar de mediante un valor único, como se ilustra en la figura 2.2.

La acción *prohibido-mover* se definiría sobre todo el conjunto de direcciones que llevan al obstáculo. La acción *intenta mover*, por su parte, se podría definir con un conjunto de valores que, centrados en la dirección del objetivo, incluyesen –con grado de preferencia decreciente– todas las direcciones hacia los lados del objetivo, $\text{objetivo} \pm \pi/2$

Mediante la definición de las acciones por medio de rangos de valores, en el momento de la decisión para poner las acciones en común, y una vez aplicada la precedencia de *prohibido-mover* sobre *intenta-mover*, seguirán estando disponibles como direcciones válidas de movimiento todas aquellas que no llevan directamente hacia el obstáculo, pero que sí acercan parcialmente al robot hacia el objetivo.

2.2.3. Modelado del mundo

Los dos aspectos previos contemplan cómo van a ser descritas las acciones y cómo las diferentes acciones van a ser puestas en común para finalmente mover el robot.

Sin embargo, para poder tomar cualquier tipo de decisión, el robot debe evaluar antes el estado de su entorno y el mundo en el que se mueve. Para esto,

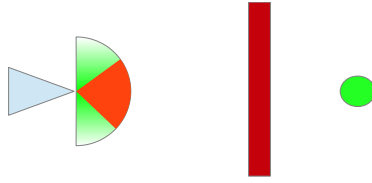


Figura 2.2: Robot con un obstáculo al frente. Muestra de la expresión de acciones de distinto tipo mediante regiones

el mundo, el entorno del robot, debe estar descrito de tal forma que pueda ser evaluado por las diferentes subtareas.

Siendo el área de aplicación de esta Tesis la navegación del robot, los elementos principales que intervienen para evaluar el entorno del robot son las posiciones relativas de los distintos objetos presentes en el mundo.

Evaluar cada una de las subtareas sobre todos y cada uno de los elementos presentes en el mundo, es una forma de proceder válida que no entra en conflicto con ninguna de las ideas expuestas previamente. Sin embargo, a medida que el mundo en el que se mueve el robot aumenta en complejidad –hay mayor número de objetos– el esfuerzo de analizar cada subtarea sobre cada objeto crecerá, pero de todos los objetos sólo unos pocos serán realmente relevantes para la navegación del robot.

Para reducir el esfuerzo de evaluar cada subtarea contra cada objeto presente en el mundo, en el método propuesto se hace uso de múltiples potenciales virtuales, con el fin de condensar la información del mundo manejada por el robot.

Agrupando la información mediante la construcción de estos potenciales virtuales, es necesaria una única evaluación de cada subtarea para considerar el total del mundo, independientemente del número de elementos presentes.

Se emplean múltiples campos de potencial virtuales, en lugar de uno solo, para mantener independientes las diferentes características necesarias para evaluar las distintas subtareas. De esta forma la subtarea encargada de evitar la colisión con obstáculos hará uso de un campo de potencial virtual integrado únicamente por obstáculos, y que no incluirá ninguna contribución del objetivo o de los otros robots presentes en el mundo.

Cada uno de los potenciales virtuales está construido para reflejar la influencia de los elementos más relevantes de aquellos que integra, para mejor ajustarse a las necesidades de la evaluación de la subtarea que lo emplea. Así, el potencial virtual de obstáculos está construido de forma que los elementos más influyentes son los obstáculos más próximos al robot, mientras que los obstáculos lejanos apenas si influyen.

Usando los tres elementos principales aquí descritos –aumento en las dimensiones del espacio de acciones, descripción de las acciones mediante regiones de

valores y uso de múltiples potenciales virtuales– se construye el método propuesto y se lleva a cabo su implementación para guiar la navegación de uno o varios robots.

2.3. Casos de estudio

Los casos elegidos para probar el método propuesto han sido seleccionados para mostrar las distintas capacidades del método y su puesta en práctica según aumenta la complejidad del caso estudiado, como puede verse en las imágenes 2.3 a 2.7.

El primero de los casos de estudio es el escenario ya descrito de un único robot que navega hacia un objetivo en presencia de un obstáculo. Este primer caso muestra cómo se evalúa la situación del robot y cómo establecer las acciones que debe seguir el robot en cada caso. Este escenario también muestra como el principal problema que surge al hacer uso de campos de potenciales, los mínimos locales, no aparece debido a la forma en que se emplean los potenciales virtuales en el método propuesto.

El segundo caso estudiado implica la navegación en el mismo espacio de varios robots, cada uno con un objetivo propio, siendo todos guiados con el mismo algoritmo. En este caso se ilustra cómo se puede ampliar el primer caso estudiado para incluir un elemento nuevo –la presencia de múltiples robots– y cómo esto afecta a la disposición de acciones y la evaluación del entorno del robot. Mediante el método propuesto se muestra como las nuevas consideraciones del escenario pueden añadirse sobre las ya existentes con una mínima variación en ellas. Así mismo, en este escenario se muestran cómo tienen lugar y son resueltas situaciones más conflictivas entre las distintas acciones que debe tomar el robot para cumplir su tarea.

El tercer escenario aumenta aún más la complejidad del conjunto de subtareas que guían el robot –sin necesidad de aumentar la complejidad del mundo–. Este tercer caso de estudio también muestra cómo la información del mundo se puede agrupar de formas distintas para propósitos distintos –el uso del mismo conjunto de fuentes de potencial, agrupadas en forma distinta, para realzar características distintas–

El cuarto caso estudiado muestra como aplicar el método propuesto a la navegación en formación de un grupo de robots mediante un conjunto simple de acciones.

A lo largo de este caso de estudio queda demostrada la capacidad de mezclar distintas acciones simples para conseguir comportamientos complejos.

Para sustentar esta hipótesis, se realiza una extensa labor de prueba con formaciones de distintas formas, distinto número de robots en la formación y distintas condiciones iniciales de los robots.

Además, después de hacer un estudio en profundidad mediante simulación, un conjunto reducido –pero aún significativo– de casos es llevado a cabo usando

robots reales. De esta forma se demuestra la aplicabilidad del método propuesto en sistemas reales y se validan los resultados obtenidos mediante simulación.

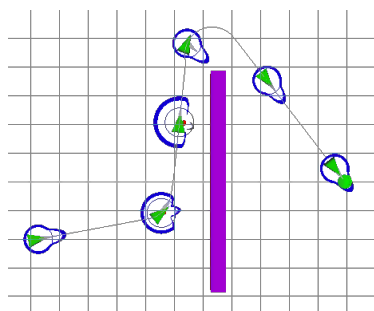


Figura 2.3: Muestra del escenario con un único robot

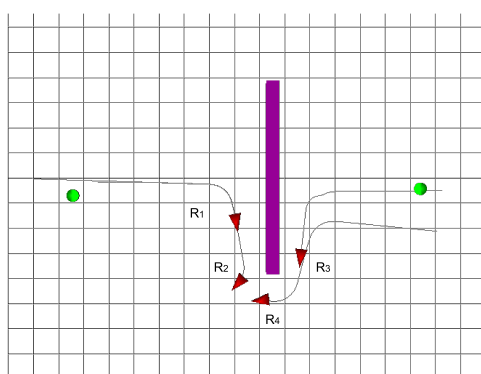


Figura 2.4: Muestra del escenario con varios robots y objetivos

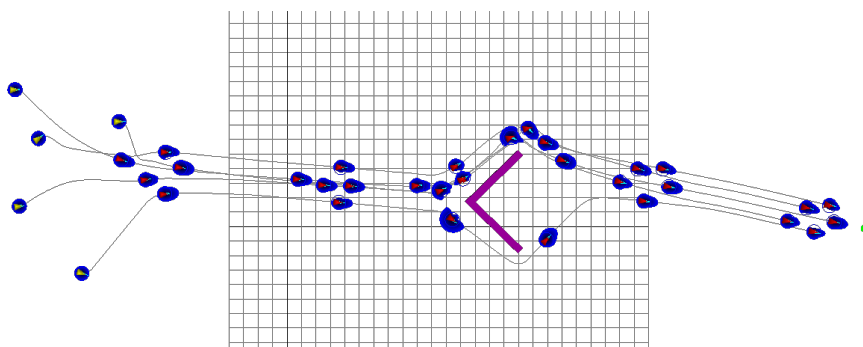


Figura 2.5: Muestra del escenario con varios robots moviéndose en grupo

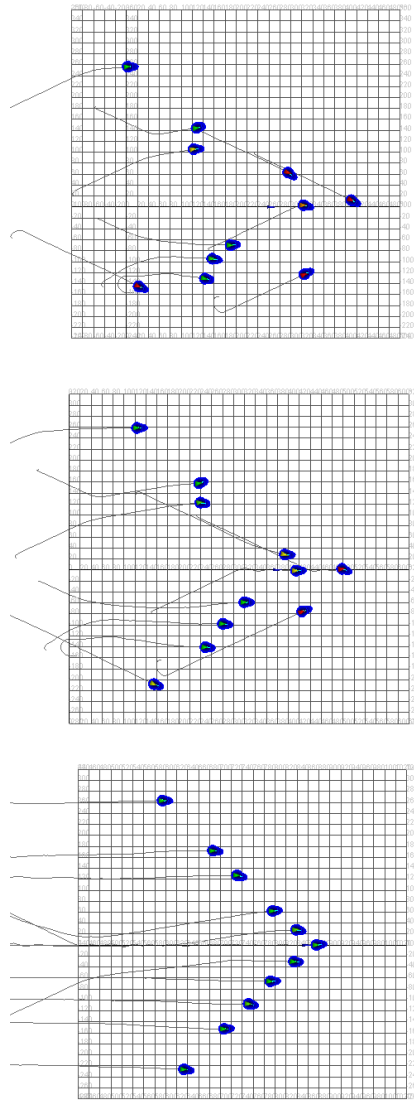


Figura 2.6: Ejemplo del proceso de construcción de una estructura de formación -cuña- para 11 robots en simulación

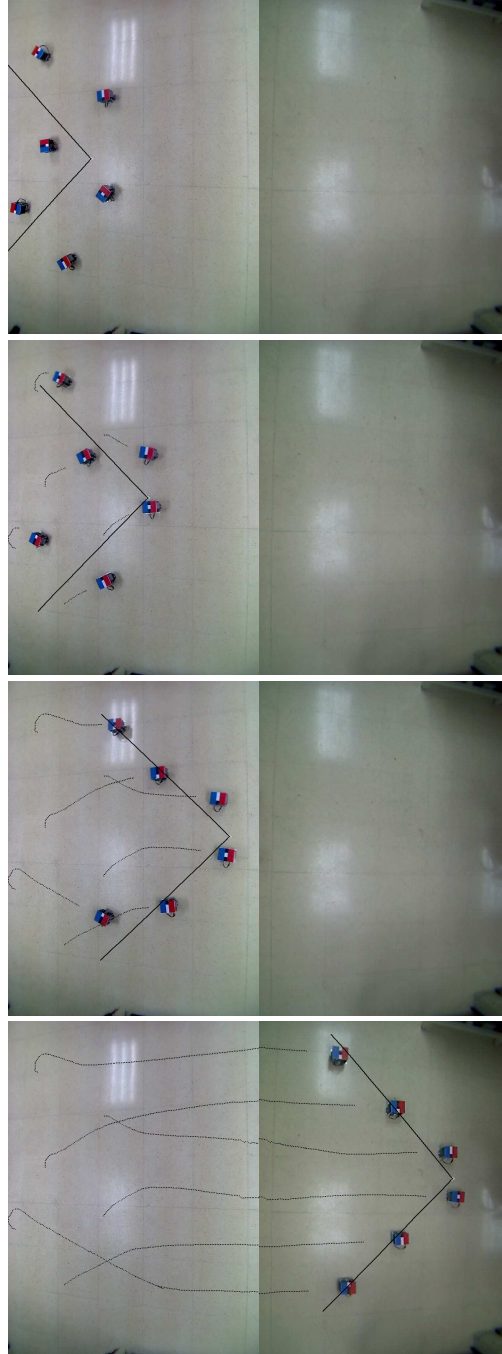


Figura 2.7: Ejemplo del proceso de construcción de una formación de cuña con 6 robots reales

2.4. Resultados

Los primeros casos de estudio se emplean principalmente para mostrar el funcionamiento del método propuesto y sus resultados, si bien satisfactorios, no son testados en profundidad. Es en el estudio del último caso, la navegación en formación, donde se lleva a cabo un análisis detallado, mostrando unos resultados satisfactorios que apoyan las diferentes hipótesis planteadas y la viabilidad del método propuesto.

Tras la simulación de la navegación en formación para distinto número de robots, se observa que la evolución de la distancia a la formación de cada robot depende principalmente del número de robots en la formación y además que esta dependencia es lineal, mientras que otros factores, como la forma de la formación sólo influyen de forma reducida. Si bien este resultado puede ser mas o menos intuitivo, si el método propuesto no fuese capaz de solucionar los conflictos a medida que la complejidad aumenta, la relación obtenida no sería lineal, dado que la complejidad del escenario en conjunto aumenta de forma geométrica con el número de robots presente.

La dependencia lineal con el número de robots presentes en el escenario se observa en todos los estadísticos analizados, apoyando así la hipótesis de que mediante el enfoque basado en comportamientos es posible distribuir la complejidad del escenario entre sus elementos, así como demuestra que el método propuesto es capaz de manejar el incremento en complejidad sin incrementar el esfuerzo necesario para su análisis.

2.5. Publicaciones

Según se han ido desarrollando y probando los diferentes elementos de la Tesis, cuando ha surgido la ocasión se ha publicado el contenido de tales avances en congresos o revistas internacionales para observar la opinión de expertos ajenos al entorno de la Tesis respecto a los avances e iniciar y poder iniciar discusión constructiva acerca del conjunto de la idea.

La primera de estas publicaciones, [Cifuentes06], versa sobre el entorno de simulación desarrollado y los elementos principales de su arquitectura.

El uso particular que se le da a los campos de potenciales en el método propuesto parte de los trabajos presentados en [Cifuentes08a] y [Cifuentes08b] donde se introduce el uso de múltiples campos de potencial para el guiado de un solo robot y un grupo de robots.

Si bien en [Cifuentes08a] y [Cifuentes08b] ya se hace uso de un conjunto de acciones superior al espacio de actuación del robot no es hasta [Cifuentes10] donde se comienza a resaltar este hecho, al describir el sistema de combinación de acciones para la navegación de un grupo de robots en formación.

Una descripción mas completa del conjunto de elementos empleados en el método propuesto puede encontrarse en [Cifuentes12a] y [Cifuentes12b] donde en dos artículos consecutivos se detalla primero el método general de uso y combinación de distintos espacios de acción hacia el espacio de actuación, que se respalda junto con el ejemplo de aplicación de un robot y un obstáculo, y, en el segundo artículo, se hace énfasis en la capacidad del método propuesto para combinar un número grande de acciones en el guiado de robot en formación.

Estos artículos son parte de la evolución de trabajo presentado en la Tesis y aunque las ideas se han mantenido a lo largo del tiempo, los detalles de la implementación y la generalidad de los métodos empleados se ha ido revisando en cada nuevo paso hasta alcanzar el documento final de la Tesis.

Capítulo 3

Detalles del Método

3.1. Robots Móviles

Durante el desarrollo del trabajo se considera la aplicación del método en robot no holónomos. Los robots empleados tienen su movimiento restringido similar al que tendría un coche: pueden girar, pero con un radio mínimo de giro y siempre y cuando, estén avanzando. La restricción a la que están sometidos los robots queda expresada en la eq. 3.1, donde $\dot{\theta}$ es la velocidad angular del robot, ν es la velocidad instantánea y r_{Lim} representa el radio mínimo con el que puede girar el robot.

$$|\dot{\theta}| \leq |\nu| / r_{Lim} \quad (3.1)$$

El valor de r_{Lim} se emplea a lo largo del método como referencia a los límites de maniobra de que es capaz el robot, junto con el valor de la velocidad máxima alcanzable por el robot, ν_{MAX} . Además, en general, se considera que el tamaño del cuerpo del robot es del orden de r_{Lim} .

Cuando ha sido necesario, se han tomado los valores de los robots experimentales como referencia de magnitud. Estos valores son : $r_{Lim} = 0,1m$ para el radio de giro y $\nu_{MAX} = 0,1m/s$ como velocidad máxima alcanzable por el robot.

En los robots empleados se supone la existencia de un sistema de control interno de velocidad y dirección de movimiento. Bajo esta consideración, el método propuesto expresa las actuaciones sobre el robot en forma de la velocidad y el rumbo deseados, pero no comprueba ni corrige su correcta aplicación.

3.2. Elementos del Algoritmo

Para la implementación del método propuesto, el algoritmo empleado hace uso de varias estructuras y construcciones previas de los datos del mundo.

3.2.1. Campos de Potencial

La descripción del mundo está realizada mediante campos de potencial virtuales. Usando estos campos se puede obtener de forma sencilla una visualiza-

ción de los distintos elementos del mundo que en otros casos podría ser difícil de realizar.

Los campos virtuales empleados son una composición de campos de potencial centrales que pueden ser expresados mediante el gradiente negativo de un potencial. Las ecuaciones 3.2 y 3.3 representan la descripción general de una fuente de potencial junto con el campo asociado, y la construcción del potencial y el campo por superposición de múltiples fuentes.

$$\phi_i(r) = a_i r^{p_i} \rightarrow \vec{f}_i(r) = -\nabla \phi_i(r) \quad (3.2)$$

$$\Phi = \sum_i \phi_i \rightarrow \vec{F} = \sum_i \vec{f}_i = \sum_i -\nabla \phi_i \quad (3.3)$$

Uso propuesto de los campos virtuales de potencial A diferencia de los métodos clásicos, donde el potencial se obtiene de la superposición de todas las fuentes existentes en el mundo –dando lugar a un campo de potencial único–, en el método propuesto se plantea la creación de varios campos virtuales.

Los campos virtuales de potencial empleados en esta Tesis agrupan las fuentes de potencial según los requisitos necesarios por la tarea a realizar. Se puede considerar entonces que existen varios tipos de potencial empleados, cada uno de ellos específico en su uso.

Otra diferencia notable con los métodos clásicos es que las fuentes de potencial, virtuales al igual que los campos, están asociadas a cada elemento presente en el mundo, pero un mismo elemento puede tener asociadas más de una fuente de potencial, de distinto carácter y que, por tanto, no se van a combinar. El uso de varias fuentes de potencial asociadas a un mismo elemento del mundo permite construir potenciales cuya evaluación final, tras la superposición de todas las fuentes específicas, represente información diferente.

Otra ventaja del empleo de múltiples campos, cada uno asociado a un objetivo concreto, es que todas las fuentes de potencial de un campo virtual específico son del mismo tipo –p.e. no se mezclan atractores con repulsores– con lo que la forma esperada tras la combinación de todas sus fuentes resulta mucho más intuitiva y se evita, en gran medida, la aparición de mínimos locales de potencial.

En general, las fuentes puntuales de potencial empleadas a lo largo del trabajo siguen la forma mostrada en las ecuaciones 3.4 y 3.5. En estas ecuaciones se puede observar la ausencia de la constante de escala a_i de la ecuación 3.2.

No es necesario el uso de la constante de escala ya que al combinarse únicamente fuentes del mismo tipo, las constantes de escala serían iguales para todos los elementos, por tanto se han obviado y reducido a la unidad.

El peso específico de un aspecto determinado se estipula al construir la evaluación de ese aspecto, con lo que aísla ese factor de la construcción de los campos en sí.

■ *Potencial Atractivo*

$$\phi_{ATT_i}(d_i) = d_i^2 \quad (3.4)$$

■ *Potencial Repulsivo*

$$\phi_{REP_i}(d_i) = 1/d_i^2 \quad (3.5)$$

3.2.2. Percepción y manejo de datos

El método de navegación propuesto es un método de carácter reactivo. El robot, a cada instante, evalúa el conjunto de elementos que le rodea y toma una decisión para su movimiento inmediatamente siguiente. Después de esto, vuelve a evaluar su entorno y tomar una nueva decisión.

El conjunto de elementos que rodea al robot en un instante dado da lugar a una colección de circunstancias que caracteriza el estado del robot.

Cada una de estas circunstancias describe un aspecto concreto del estado del robot que surge de combinar el estado de uno o varios elementos que rodean al robot. Cada una de estas circunstancias es denominada *Situation* y cada uno los elementos evaluados para alcanzar una circunstancia es denominado *Descriptive Element*.

Elementos descriptivos, *Descriptive Elements* Un elemento descriptivo representa la evaluación del estado de un elemento concreto del mundo. Esta evaluación es continua y normalizada, normalmente en el intervalo $[0.,1]$ y en algunas ocasiones en el intervalo $[-1.,1]$. Los elementos descriptivos se construyen a partir de los diferentes valores de campo o potencial de los distintos campos virtuales empleados en cada problema.

Los elementos descriptivos se combinan para crear situaciones. Al estar normalizados en un rango continuo, los elementos descriptivos pueden operarse entre sí y el resultado se puede asociar a una lógica continua, como más o menos se describe en las ecuaciones 3.6 a 3.8.

De este modo, un producto de dos elementos descriptivos dará lugar a una relación conjuntiva –ambos elementos descriptivos deben tener un valor alto para que su combinación tenga un valor alto– y del mismo modo, se puede negar un elemento descriptivo restando este a la unidad. También se puede emplear la graduación de un elemento descriptivo mediante el uso de la potencia, así, un elemento descriptivo elevado al cuadrado sólo resultará en una situación de valor alto si el valor del elemento descriptivo es muy alto.

$$DE_1 \& DE_2 = DE_1 \times DE_2 \quad (3.6)$$

$$!DE_1 = 1 - DE_1 \quad (3.7)$$

$$”Mu” DE = DE^2 \quad (3.8)$$

Para aquellos elementos cuya evaluación esté naturalmente acotada –medidas angulares–, se hace uso tanto de transformaciones lineales como trigonométricas para su normalización.

Para la evaluación de parámetros que no tienen unos límites naturales –una distancia–, se ha empleado normalmente la función Sigmoide para la normalización, expresada en la forma de la ecuación 3.9.

$$\begin{aligned}
Sg(x, x_{001}, x_{099}) &= \frac{1}{1 + e^{-(x-o) \cdot s}} \\
o &= (x_{001} + x_{099}) / 2 \\
s &= \left(\ln \left(\frac{1}{0.01} - 1 \right) - \ln \left(\frac{1}{0.99} - 1 \right) \right) / (x_{099} - x_{001})
\end{aligned} \tag{3.9}$$

Situaciones, *Situations* Las situaciones están consideradas como la valoración de un conjunto de elementos determinados del entorno del robot que pueden asociarse con una idea global abstracta ante la que le robot debe emprender alguna acción.

Formalmente, las situaciones se construyen como una combinación de elementos descriptivos y, como éstos, están representadas por un valor continuo y normalizado en el intervalo $[0..1]$. Un ejemplo sería el ilustrado en la ecc 3.10 donde los elementos descriptivos Obs_{Cerca} y $Obs_{Delante}$ se combinan para crear una situación específica, donde Obs_{Cerca} indica la proximidad de un obstáculo –normalizando la distancia mediante una Sigmoide según la ecuación 3.11 con puntos de referencia para *cerca* y *lejos*– y donde $Obs_{Delante}$ indica la dirección en la que se encuentra el obstáculo –normalizada mediante la función de la ecuación 3.12–.

$$S_{CaminoObstruido} = Obs_{Cerca} \times Obs_{Delante} \tag{3.10}$$

$$Obs_{Cerca} = Sg(Dist_{Obs}, ref_{cerca}, ref_{lejos}) \tag{3.11}$$

$$Obs_{Delante} = \frac{1 - \cos(\theta_{Obs})}{2} \tag{3.12}$$

En el método propuesto las situaciones llevan asociadas una serie de acciones, de modo que éstas tienen más o menos peso en función de la valoración de su situación.

3.3. Proceso de decisión

El objetivo final del algoritmo propuesto es la navegación del robot cumpliendo los requisitos de su tarea. La navegación del robot se realiza mediante la definición de una serie de acciones que afectan al movimiento y, en conjunto, lo definen.

3.3.1. Descripción de las Acciones

Cada acción se define mediante un conjunto extenso de valores sobre los que la acción se aplica con distinta intensidad. Ésto permite aumentar el conjunto de soluciones viables cuando todas las acciones, tomadas de forma independiente, se ponen en común para su aplicación sobre los actuadores del robot. Para definir el conjunto de valores sobre los que aplicar una acción, a lo largo de este trabajo se emplea un función de campana extendida, ecuación 3.13, que modifica a la función clásica, ecuación 3.14, de modo que la zona de máximo de la campana pueda ser un rango amplio en lugar de un valor puntual, tal como se ilustra en la figura 3.1.

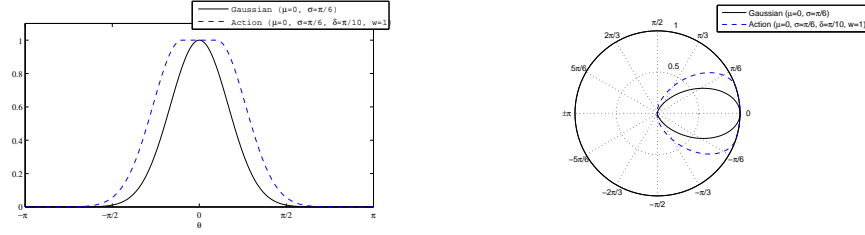


Figura 3.1: Campana clásica y campana extendida. Representaciones cartesiana y polar.

$$\tilde{\psi}(\theta; \mu_i, \sigma_i, \delta_i, w_i) = \begin{cases} \psi(\theta; \mu_i - \delta_i, \sigma_i) * w_i, & \mu_i - \pi \leq \theta \leq \mu_i - \delta_i \\ w_i, & \mu_i - \delta_i < \theta < \mu_i + \delta_i \\ \psi(\theta; \mu_i + \delta_i, \sigma_i) * w_i, & \mu_i + \pi \geq \theta \geq \mu_i + \delta_i \end{cases} \quad (3.13)$$

$$\psi(\theta; \mu, \sigma) = e^{-\frac{(\theta - \mu)^2}{2\sigma^2}} \quad (3.14)$$

3.3.2. Acciones empleadas

El movimiento real del robot se lleva a cabo mediante sus actuadores, sin embargo el método propuesto hace uso de un mayor número de acciones que actuadores hay disponibles en el robot. De esta forma, se pretende simplificar el proceso de combinar las múltiples acciones, asociadas a distintas situaciones, en su aplicación sobre los actuadores del robot.

Cada situación puede definir una o varias acciones. A lo largo del trabajo se emplean tres tipos distintos de acciones para definir el movimiento deseado del robot en base a cada situación.

- *Elect*: Este tipo de acciones permiten definir una dirección de movimiento deseada para el robot, definiendo aquellas direcciones que hacen al robot avanzar en la realización de su tarea.
- *Forbid*: Las acciones de este tipo definen aquellas direcciones que el robot debe evitar, ya sea porque suponen un peligro para el robot o porque suponen un retroceso en la realización de su tarea. Estas acciones están concebidas de forma que tengan prioridad sobre las acciones de tipo *Elect*.
- *Brake*: Mediante estas acciones es posible definir un límite a la velocidad máxima que puede alcanzar el robot en una dirección determinada. En ausencia de otra indicación el robot intentará alcanzar siempre la velocidad máxima.

3.3.3. Superposición de acciones

Las diferentes situaciones consideradas a la hora de definir el comportamiento de un robot pueden definir múltiples acciones, ya sean de tipos distintos o iguales.

La naturaleza de cada una de las acciones definida permite agrupar las diferentes acciones del mismo tipo definidas por todas las situaciones consideradas. El conjunto de acciones de un mismo tipo se denomina a lo largo de este trabajo como Mapa de acciones, *Action Map*.

Existen por tanto tres mapas de acciones, uno por cada tipo de acción:

- *Elected*: Representa el conjunto de direcciones de movimiento deseables para el robot.
- *Frobidden* : Representa el conjunto de direcciones desaconsejables para el robot, sin importar lo deseables que sean.
- *Braked*: Representa un mapa de las velocidades máximas permisibles en cada dirección posible de movimiento del robot.

La diferente naturaleza de las acciones y mapas de acciones propuestos, aconseja que la agrupación de acciones para la construcción de su correspondiente mapa sea de carácter particular.

De este modo, las acciones de tipo *Elect*, al representar el nivel de bondad de una dirección y el beneficio observado por cada situación de moverse en una dirección concreta, dan lugar a un mapa de acción que se construye de forma aditiva como representa la ecuación 3.15 donde $E(\theta; \mu_i, \sigma_i, \delta_i, w_i)$ representa a cada acción tipo *Elect* individual tomada a lo largo de la valoración de todas las situaciones.

$$Elected(\theta) = \sum E(\theta; \mu_i, \sigma_i, \delta_i, w_i) \quad (3.15)$$

Por otro lado, las acciones de tipo *Forbid* y *Brake*, de naturaleza restrictiva, deben combinarse de modo que al final se aplique la máxima restricción al robot para cada posible dirección de movimiento. Para ello las acciones de este tipo se combinan mediante una unión por superposición, tal como se expresa en las ecuaciones 3.16 y 3.17.

Los diferentes modos de combinación se ilustran en la figura 3.2

$$Forbidden(\theta) = \bigcup [F(\theta; \mu_1, \sigma_1, \delta_1, w_1), \dots, F(\theta; \mu_i, \sigma_i, \delta_i, w_i), \dots] \quad (3.16)$$

$$Braked(\theta) = \bigcup [B(\theta; \mu_1, \sigma_1, \delta_1, w_1), \dots, B(\theta; \mu_i, \sigma_i, \delta_i, w_i), \dots] \quad (3.17)$$

3.3.4. Procesado de acciones y composición final

El procesado de los diferentes mapas de acciones está altamente relacionado con las capacidades de actuación del robot. En el caso del trabajo realizado, se trata de aplicar el algoritmo propuesto en robots no holónomos, y por tanto las restricciones a las que están sometidos deben tenerse en cuenta.

Para la decisión final sobre cómo mover el robot, se evalúan los mapas de acciones para considerar los diversos factores que deben ser tenidos en cuenta. Para mantener la máxima cantidad de información a lo largo de este proceso, las evaluaciones intermedias se llevan acabo mediante mapas de valoración, similares a los mapas de acción.

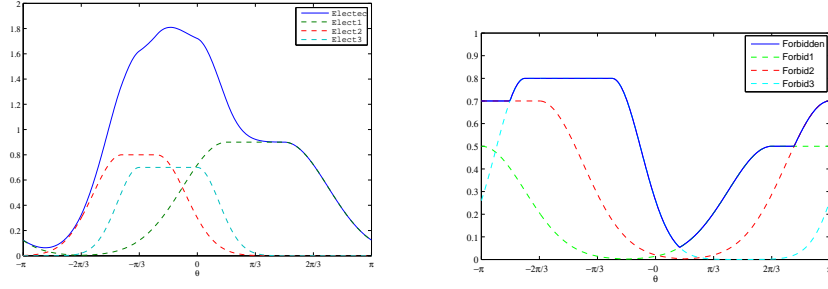


Figura 3.2: Mapas de acciones de tipo *Elected* y tipo *Forbidden* a partir de varias contribuciones individuales

- El primero de los mapas de valoración es el mapa de viabilidad, *Suitability Map*, este mapa determina el interés relativo de cada dirección. Para ello se emplean los mapas de acción *Elected* y *Forbidden* siguiendo la ecuación 3.18, cuyo resultado se puede apreciar en la figura 3.3. En este mapa, un valor $S(\theta) = 1$ implica una valoración neutral de la dirección –viable pero no deseada–, $S(\theta) = 2$ indica una dirección deseable y $S(\theta) = 0$ una dirección no viable.

$$S(\theta) = [1 - \text{Forbidden}(\theta)] \times \left(1 + \frac{\text{Elected}(\theta)}{\max(\text{Elected})} \right) \quad (3.18)$$

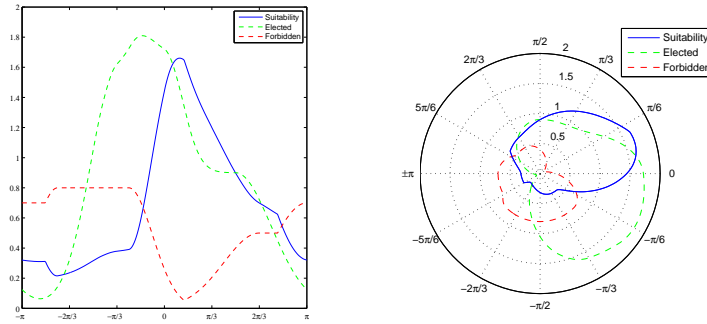


Figura 3.3: Ejemplo de la construcción del mapa de viabilidad

- Debido a la naturaleza no holónoma de los robots empleados, es necesaria la construcción de un mapa de seguridad, *Safety Map*, que refleja, no sólo la idoneidad de la dirección final, sino también el paso por zonas no deseadas a lo largo de la maniobra de giro desde la dirección actual del robot hasta la dirección final evaluada. Este mapa debe realizarse por duplicado, uno para giro horario y otro anti-horario, donde cada uno se construye mediante el algoritmo descrito en el pseudo-código de la ecuación 3.19 cuyo resultado se ilustra en la figura 3.4.

$$\begin{aligned}
M &= 1 - \min (Forbidden(\xi)); \quad 0 \leq \xi \leq 2\pi \\
L_0 &= 1 \\
FOR \quad \beta &= 0 : s : 2\pi \\
L_1 &= 1 - q \cdot [Forbidden(\beta) - \min (Forbidden(\chi))]; \quad 0 \leq \chi \leq \beta \\
L_0 &= L_0 \cdot L_1 \\
SF(\beta) &= L_0 \cdot \frac{1 - \min (Forbidden(\chi))}{M}
\end{aligned} \tag{3.19}$$

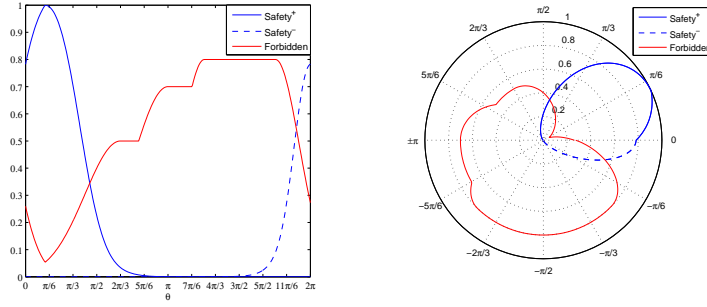


Figura 3.4: Ejemplo de la construcción del mapa de seguridad

- Finalmente, se construye un mapa de giro, *Turning Map*, que combina los resultados de la viabilidad de cada dirección, la seguridad para alcanzar dicha dirección como final y el coste implicado en el proceso de giro. Este mapa, de nuevo doble, se construye siguiendo la ecuación 3.20, y su resultado se ilustra en la figura 3.5

$$TM^{\pm}(\theta) = \left(S(\theta) - w_t \left| \frac{\theta}{2\pi} \right| \right) \cdot SF^{\pm}(\theta) \tag{3.20}$$

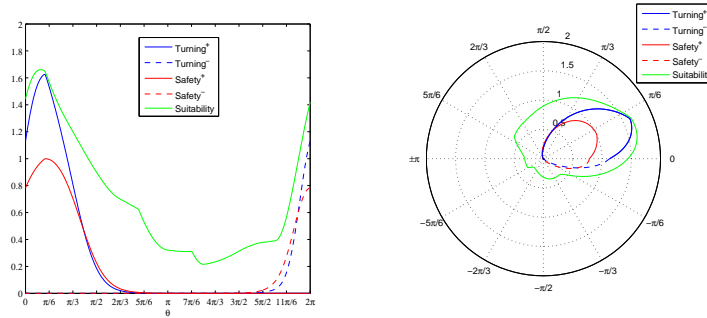


Figura 3.5: Ejemplo de la construcción del mapa de giro

Construidos los mapas de valoración, la dirección final del robot se establece como aquella dirección con un mayor valor en el mapa de giro, como se especifica en la ecuación 3.21

$$\theta_{FINAL} : TM(\theta_{FINAL}) = \text{máx}(TM^{\pm}(\theta)) \quad (3.21)$$

Para establecer la velocidad máxima del robot, se emplea el valor del mapa de frenado en la dirección instantánea del robot, tal como se establece en la ecuación 3.22

$$\nu_{ROBOT} = \nu_{MAX} \times (1 - Braked(\theta = 0)) \quad (3.22)$$

Capítulo 4

Casos de Estudio

4.1. Robot y Obstáculo

Para estudiar el caso simple de un único robot que navega hacia un objetivo y encuentra un obstáculo en su camino necesitamos simplemente la información del objetivo y el obstáculo. Para ésto definimos dos tipos de fuentes de potencial, ecuaciones 4.2 y 4.1, uno para el obstáculo $-\phi_{OBS}$ y otro para el objetivo $-\phi_{GOAL}$, y asociamos a cada elemento su fuente.

$$\phi_{GOAL}(d) = d^2 \quad (4.1)$$

$$\phi_{OBS}(d) = 1/d^2 \quad (4.2)$$

Mediante estos campos podemos definir los elementos descriptivos necesarios.

Por un lado se define el elemento descriptivo asociado a la proximidad del obstáculo, ecuación 4.3. Esta ecuación hace uso del valor definido en la ecuación 4.4 para establecer la distancia del obstáculo a partir del campo de potencial de obstáculo. Los factores de las ecuaciones 4.5 y 4.6 establecen qué se considera cerca y qué se considera lejos –definido en función del límite de maniobra del robot, r_{Lim} –

$$\begin{aligned} Obs_{Near} &= Sg(x; x_{001}, x_{099}) \\ x &= Obs_{Distance} \\ x_{001} &= d_{Far} \\ x_{099} &= d_{Close} \end{aligned} \quad (4.3)$$

$$Obs_{Distance} = \frac{1}{\sqrt{\phi_{Obs}}} \quad (4.4)$$

$$d_{Close} = \left(1 + \left|\frac{\theta_{OBS}}{\pi}\right|\right) \cdot r_{Lim} \quad (4.5)$$

$$d_{Far} = 3 \cdot d_{Close} \quad (4.6)$$

Por otro lado se define el elemento descriptivo asociado a indicar cuándo el obstáculo está en el camino del objetivo, ecuación 4.7, donde se comparan los ángulos de los campos de obstáculo y de objetivo.

$$Obs_{InGoalPath} = \frac{1 - \cos(\theta_{OBS} - \theta_{GOAL})}{2} \quad (4.7)$$

Definidos los elementos descriptivos es posible definir las situaciones. La situación $S_{1,0}$ indica que el camino hacia el objetivo está libre, mientras que $S_{1,1}$ indica que hay un obstáculo en el camino hacia el objetivo.

$$S_{1,0} = 1 - (Obs_{Near} \cdot Obs_{InGoalPath}) \quad (4.8)$$

$$S_{1,1} = Obs_{Near} \cdot Obs_{InGoalPath} \quad (4.9)$$

Para definir la actuación del robot en cada momento, se definen una serie de acciones asociadas a las dos posibles situaciones.

En la tabla 4.1 se definen las acciones que ha de tomar el robot asociadas con la situación $S_{1,0}$. En este caso, las acciones se limitan a una única acción de tipo *Elect* para indicar al robot que es deseable el movimiento en la dirección del objetivo y, en menor grado, todas las direcciones que acerquen parcialmente al robot hacia el objetivo.

Cuadro 4.1: Acciones asociadas a la situación $S_{1,0}$

Acción	μ	σ	δ	w
<i>Elect</i> (μ, σ, δ, w)	θ_{GOAL}	$\pi/4$	0	$(S_{1,0})^2$

En la tabla 4.2 están las acciones asociadas a la situación $S_{1,1}$. En este caso se trata de tres acciones: Una acción de tipo *Forbid* que indica al robot que restrinja el movimiento en la dirección del obstáculo y, en menor grado, de la zona entorno a él. La segunda acción, de tipo *Elect* indica al robot que se mueva en la dirección opuesta al obstáculo o cualquiera que se aleje parcialmente de él. En la tercera acción, de tipo *Brake*, se define una reducción en la velocidad para aquellas direcciones que acerquen al robot hacia el obstáculo.

Cuadro 4.2: Acciones asociadas a la situación $S_{1,1}$

Acción	μ	σ	δ	w
<i>Forbid</i> (μ, σ, δ, w)	$\theta_{OBS} + \pi$	$(\pi/2) \cdot S_{1,1}$	$(\pi/2) \cdot S_{1,1}$	$S_{1,1}$
<i>Elect</i> (μ, σ, δ, w)	θ_{OBS}	$\pi/4$	$\pi \cdot (1 - S_{1,1})$	$S_{1,1}$
<i>Brake</i> (μ, σ, δ, w)	$\theta_{OBS} + \pi$	$(\pi/3) \cdot S_{1,1}$	$(\pi/3) \cdot S_{1,1}$	$S_{1,1}$

Es necesario señalar que todas las acciones van a ser aplicadas en cada ciclo de control del robot, pero el peso de cada acción está ligado a la evaluación de su situación por la expresión indicada en el termino w de las anteriores tablas.

El resultado final del movimiento del robot en un escenario simulado regido por este conjunto de situaciones y acciones se puede observar en la figura 4.1, así como la evolución de los situaciones planteadas, en la figura 4.2

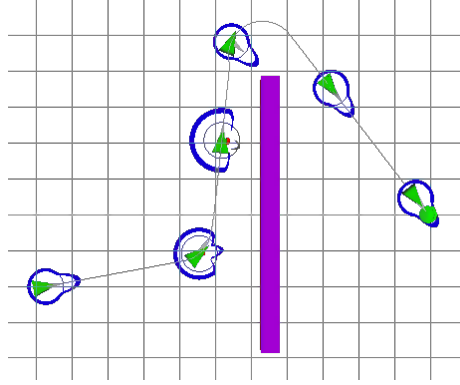


Figura 4.1: Recorrido del robot en su movimiento hacia el objetivo. En cada captura se muestra entorno al robot el mapa de viabilidad; el círculo fino representa la unidad.

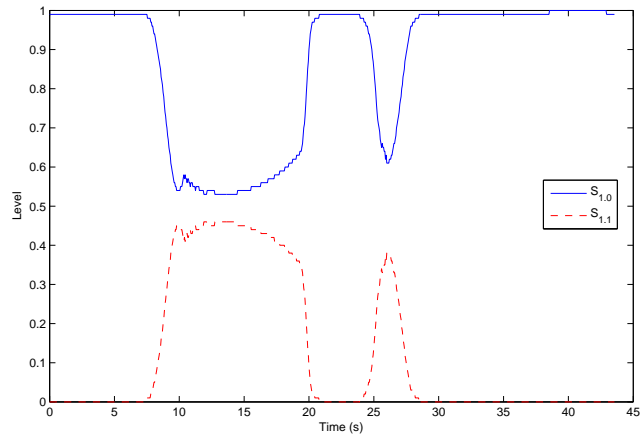


Figura 4.2: Evolución de los indicadores de situación $S_{1,0}$ (continua) y $S_{1,1}$ (punteada) en el robot a lo largo de la simulación

4.2. Navegación en formación

4.2.1. Campos

Para la navegación en formación de un grupo de robots es necesario conocer una mayor cantidad de información: la estructura de la formación, la dirección de movimiento del grupo, la presencia de obstáculos y la de los otros robots de la formación.

Siguiendo el método planteado, la información del mundo se agrupa en tres tipos distintos de potenciales : obstáculos, robots y formación, mientras que la dirección de movimiento de la formación, θ_{Lead} , se transmite a cada robot integrante del grupo o es obtenida a partir de otros elementos del sistema.

El campo de potencial ligado a los obstáculos es el mismo potencial de repulsión empleado en el caso del robot único, y el campo empleado por los robots tiene la misma expresión, si bien se mantiene separado.

El campo de potencial de la formación se construye de modo que en cada punto del espacio indique al robot la dirección en la que se encuentra más próxima la estructura de la formación –línea de traza o punto de situación–, donde el potencial sigue una forma similar a la de un atractor puntual como el usado en el objetivo del caso del robot único.

Estrategia de Formación La estructura de formación estudiada puede definirse mediante un traza continua, recta o curva, cerrada o abierta, o mediante una colección de puntos sobre los que deben situarse los robots del grupo. En general, el aspecto del potencial de formación se considera equivalente a situar un atractor puntual en el punto mas próximo de la estructura al robot.

Un aspecto a resaltar es el que cada robot actúa de forma completamente independiente de los otros, no hay comunicación explícita.

Cada robot, conociendo la estructura de la formación requerida y la posición de los otros robots del grupo, ajusta la posición de la estructura de la formación sobre el conjunto de los robots de modo que se minimice la suma del campo de formación de todos los robots del grupo, incluido él mismo. Una vez que el robot sitúa la estructura de formación sobre el conjunto de robots del grupo, hace uso de su propia lectura de campo y potencial de formación para reducir su propio valor de campo y así situarse mejor en el conjunto de la formación.

4.2.2. Elementos descriptivos, Situaciones y Acciones

Obstáculos Los obstáculos se tratan de una forma similar a como son empleados en el caso del robot único, la principal diferencia está en que al no haber un objetivo definido, simplemente se observa la aparición de obstáculos por delante del robot. De esta forma se define un elemento descriptivo Obs_{InPath} que indica la presencia de obstáculos por delante del robot y otro elemento descriptivo Obs_{Near} igual al empleado en 4.4.

Con estos elementos se define la situación que vigila la proximidad con obstáculos, $S_{1,1}$, ecuación 4.10. Las acciones que aplica el robot para evitar la colisión con los obstáculos son las mismas que en el caso del robot único, situadas en la tabla 4.2.

$$S_{1,1} = Obs_{Near} \cdot Obs_{InPath} \quad (4.10)$$

Otros robots El campo de potencial de los robots está principalmente dirigido a evitar colisiones entre robots.

La evaluación de la proximidad de los otros robots se refleja en el elemento descriptivo Rob_{Near} . El espacio de colisión es considerado como la zona donde el robot debe comenzar las acciones para evitar la colisión con otro robot.

Otros elementos descriptivos se definen para reflejar la posición relativa de los otros robots: Rob_{Ahead} indica la presencia de robots por delante, Rob_{Back} indica la presencia de robots por detrás y $Rob_{SaddlePoint}$ indica que el robot –el que evalúa la situación– tiene a varios robots a su alrededor.

Mediante estos elementos descriptivos se construye la situación que evalúa la presencia de otros robots para evitar la colisión con ellos, ecuación 4.11. Así mismo, las acciones que debe emprender el robot para evitar la colisión, tabla 4.3, consisten en evitar la dirección de los otros robots mediante una acción de tipo *Forbid* y, por si acaso, establecer una velocidad reducida en la dirección de esos otros robots mediante una acción de tipo *Brake*.

$$S_{2,1} = Rob_{Near} \cdot Rob_{Ahead} \cdot (1 - Rob_{SaddlePoint}) \quad (4.11)$$

Cuadro 4.3: Acciones relativas a la situación $S_{2,1}$

Acción	μ	σ	δ	w
<i>Forbid</i> (μ, σ, δ, w)	$\theta_{ROB} + \pi$	$(\pi/2) \cdot S_{2,1}$	$(\pi/3) \cdot S_{2,1}$	$S_{2,1}$
<i>Brake</i> (μ, σ, δ, w)	$\theta_{ROB} + \pi$	$(\pi/3) \cdot S_{2,1}$	$(\pi/6) \cdot S_{2,1}$	$S_{2,1}$

Formación Para alcanzar y mantener la estructura de formación es necesario conocer tanto la relación de la posición del robot respecto a la formación, como la relación de los otros robots con la estructura de la formación. Estas relaciones, como siempre, se definen mediante elementos descriptivos.

La posición del robot respecto a la estructura de formación se describe mediante O_{AheadF} , O_{BackF} , O_{RightF} , O_{LeftF} , que indican si el robot está al frente, a la espalda, a la derecha o a la izquierda de la formación, tomando como referencia la dirección de movimiento de la formación y comparándola con la lectura del campo de formación. La construcción de estos elementos descriptivos se hace de forma similar a como se ha construido $Obs_{InGoalPath}$ en el caso del robot único. Así mismo, se define O_{FSide} para indicar si el robot está situado completamente a un lado de la formación–cualquiera de ellos–.

Para la distancia del robot a la formación se define otro elemento descriptivo, D_{FI} mediante una función sigmoide de la misma forma que se definió Obs_{Near} . Este elemento descriptivo establece las zonas dónde el robot se considera lejos de la formación, $D_{FI} \approx 1$, –y ha de dirigirse hacia ella–, dónde se considera dentro de la formación, $D_{FI} \approx 0$, –y ha de mantener la estructura–.

y la zona intermedia, $0 < D_{FI} \approx 1$ –donde debe maniobrar para alcanzar un lugar en la estructura.

También es necesario conocer la posición relativa de los otros robots con respecto de la dirección de movimiento de la formación para mejor adaptar la estrategia de movimiento del robot.

A este efecto se definen O_{AheadR} , O_{BackR} y O_{RSide} que indican cuando los otros robots están situados por delante, por detrás o a un lado, siendo "delante" la dirección de movimiento de la formación.

Así mismo, se define un nuevo espacio de distancia con el elemento descriptivo D_{RM} . Este elemento descriptivo refleja la disposición de los otros robots respecto a un espacio enfocado a las maniobras entre robots. El espacio de maniobra, más lejano que el de colisión, es la región donde el robot empieza a maniobrar para moverse entorno a otros robots que buscan, o ya han alcanzado, una posición en la estructura de formación. Este elemento descriptivo se construye haciendo uso del campo de los otros robots.

Las situaciones que se han definido para agrupar las acciones de los robots con relación a la formación son dos: $S_{3,1}$, ecuación 4.12, que indica cuándo el robot está lejos de la formación –y debe acercarse a ella– y $S_{3,2}$, ecuación 4.13, que indica cuándo el robot está cerca de la formación –y por tanto debe ajustarse y mantener la estructura–.

$$S_{3,1} = D_{FI} \quad (4.12)$$

$$S_{3,2} = (1 - D_{FI}) \quad (4.13)$$

Las acciones asociadas a la situación $S_{3,1}$, orientadas a guiar al robot hacia la formación, comprenden dos grupos de acciones.

El primer grupo, tabla 4.4 guía al robot hacia la formación cuando está situado lejos de ésta mediante una acción de tipo *Elect* y otra de tipo *Brake* dirigidas a mover el robot en una trayectoria de intercepción con la formación, o, si está situado muy por delante, alinear el robot con la dirección de movimiento de la formación y frenar para esperar la llegada de la formación.

El segundo grupo, en la tabla 4.5, cubre el caso especial en que se da una simetría entre dos robots que se aproximan a la formación desde lados opuestos de ésta. En ese caso la acción de tipo *Brake* frena al robot situado en la derecha de la formación con lo que da preferencia al situado en el lado izquierdo, lo que resuelve la simetría.

Cuadro 4.4: Acciones de aproximación a la formación asociadas a la situación $S_{3,1}$

Acción	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	μ_E	$\pi/2$	0	$S_{3,1}$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3,1} \cdot O_{AheadF} \cdot freePath$

Donde:

$$\mu_E = \theta_{Lead} + \pi/5 \cdot O_{FSide} \cdot freePath$$

$$freePath = (1 - O_{AlignedRF} \cdot (1 - D_{RM}))$$

Cuadro 4.5: Acción para evitar la simetría asociada a la situación $S_{3,1}$

Acción	μ	σ	δ	w
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/4$	0	$S_{3,1} \cdot O_{RightF} \cdot \sqrt{1 - freePath}$

Las acciones asociadas a $S_{3,2}$ están orientadas a mantener la estructura de formación y facilitar la inserción de otros robots por parte de aquellos que ya están dentro. De nuevo estas acciones están reunidas en dos grupos.

El primer grupo de acciones, en la tabla 4.6 se encarga de mantener a los robots dentro de la estructura de la formación. Ésto se hace mediante una acción de tipo *Elect* que alinea al robot principalmente en la dirección de movimiento del grupo con pequeñas correcciones en la dirección del campo de formación cuando el robot no se encuentra exactamente en el interior de la estructura. Junto a esta acción, para limitar la velocidad del grupo, y permitir a robots fuera de la estructura alcanzarla, se establece una segunda acción de tipo *Brake* en la dirección de avance del grupo para aquellos robots dentro de la estructura. Una acción más de tipo *Brake* se incluye en este grupo, orientada a frenar más intensamente a aquellos robots que se salgan de la estructura adelantándose a la estructura.

Cuadro 4.6: Acciones para mantener la formación asociadas a la situación $S_{3,2}$

Acción	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	μ_E	$\pi/2$	0	$S_{3,2} \cdot O_{AlignedRF} \cdot (\frac{1}{2} + \frac{1}{2} D_{RM})$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3,2} \cdot O_{BackF} \cdot D_{RM} \cdot \ \vec{\nabla} \phi_{Form}\ \cdot 0,1$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3,2} \cdot (1 - O_{BackR} \cdot (1 - D_{RM})^2) \cdot 0,2$

Donde:

$$\mu_E = \theta_{Lead} + \pi/5 \cdot O_{FSide} \cdot O_{AlignedRF} \cdot D_{FI}$$

El segundo grupo de acciones, tabla 4.7, está orientado a que aquellos robots situados dentro de la formación faciliten la inserción de los robots situados fuera, pero de forma que no se desajuste la estructura ya formada. Para ésto se hace uso de una acción de tipo *Elect*, para desplazar ligeramente al robot en la dirección opuesta del otro robot, siempre sin salirse de la zona interior de la estructura. La segunda acción, de tipo *Brake*, frena ligeramente al robot en el interior de la estructura si el otro robot está situado al frente.

Cuadro 4.7: Acciones para facilitar la inserción asociadas a la situación $S_{3,2}$

Acción	μ	σ	δ	w
$E(\mu, \sigma, \delta, w)$	μ_E	$\pi/2$	0	$S_{3,2} \cdot (1 - D_{RM}) \cdot 0,5$
$B(\mu, \sigma, \delta, w)$	θ_{Lead}	$\pi/3$	0	$S_{3,2} \cdot (1 - D_{RM}) \cdot O_{AheadR} \cdot 0,6$

Donde:

$$\mu_E = \theta_{Lead} + \pi/5 \cdot O_{RSide} \cdot D_{FI}$$

La combinación de las 8 acciones orientadas a guiar a los robots entorno a la formación, las 2 orientadas a evitar las colisiones entre miembros del grupo, y las 3 orientadas a evitar la colisión con obstáculos, permiten a un grupo de robots navegar de forma coherente y alcanzar una estructura genérica de formación, sin necesidad de más consideraciones.

Capítulo 5

Análisis de resultados

5.1. Simulación

Mediante simulación, se ha realizado el análisis sistemático de un grupo de robots siguiendo el conjunto de situaciones/acciones propuesto para la navegación en formación. En este análisis se han empleado formaciones de 3, 4, 5, 6, 7, 9, 11, 15, 20, y 30 robots, así como formaciones en línea –alineados en la dirección de movimiento–, en columna –alineados perpendiculares a la dirección de movimiento–, en cuña y en círculo. Además, para cada combinación de número de robots y estructura de formación se ha repetido la prueba con 101 posiciones y direcciones iniciales de los robots aleatorias y todo el conjunto de pruebas se han realizado con y sin inyección de ruido. Como resultado se han obtenido los detalles de la navegación de casi cien mil robots.

Estadísticos Los datos obtenidos se han analizado estudiando dos grupos principales.

El primer grupo, la evolución de la distancia de cada robot a la estructura de formación a lo largo del tiempo de simulación, permite analizar el comportamiento individual de cada robot en su interacción con el conjunto de elementos del escenario.

El segundo grupo, el tiempo necesario para alcanzar una estructura de formación estable –todos los robots dentro de la estructura y sin perturbaciones–, permite estudiar la evolución del conjunto de robots actuando como un grupo, resultado de las interacciones entre sus individuos.

En los resultados obtenidos se ha observado que la evolución de la distancia de cada robot a la estructura de formación a lo largo del tiempo evolucionaba de forma similar a un decaimiento exponencial, como se puede ver en la figura de ejemplo 5.1. Esta tendencia implica que los robots, en general, se mueven de modo que mejoran su posición respecto a la formación a lo largo del tiempo.

A la vista de esta tendencia, se han ajustado los resultados de cada conjunto de pruebas a una expresión de decaimiento exponencial con los resultados mostrados en la figura 5.2.

De igual modo, la desviación estándar de la distancia del robot a la formación, sigue el mismo patrón y ha sido sometida al mismo análisis con los

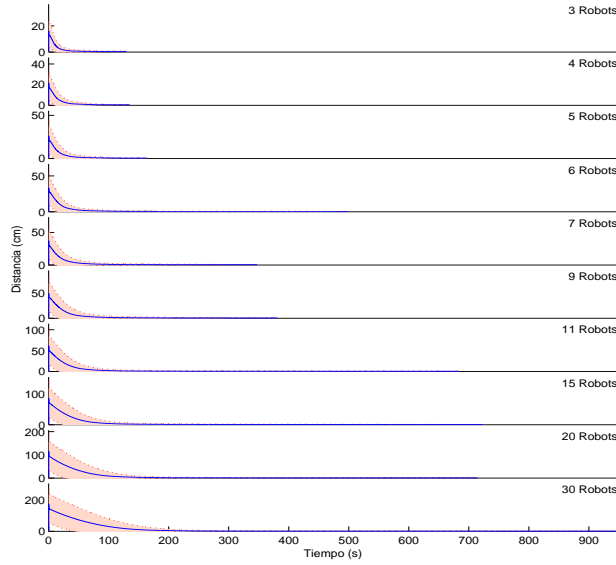


Figura 5.1: Formación en cuña, Distancia media de los robots a la formación a lo largo del tiempo de simulación \pm desviación estándar para 101 simulaciones de cada grupo

resultados mostrados en 5.3. Esta tendencia implica que los robots no sólo se distribuyen entorno a la formación, si no que la formación se vuelve más definida a medida que avanza el tiempo.

En los resultados obtenidos del tiempo necesario para alcanzar una estructura estable, el histograma muestra una cierta distribución, figura 5.4, de modo que se han estudiado el tiempo medio y la desviación para cada conjunto de ensayos con los resultados mostrados en la figura 5.5.

En los tres conjuntos de datos se puede apreciar claramente como existe una tendencia lineal del número de robots en la formación con el tiempo de decaimiento de la distancia, el tiempo de decaimiento de la desviación de la distancia y el tiempo medio de estabilización de la formación.

Dado que el número de interacciones a considerar en un escenario cualquiera está en relación con el cuadrado del número de robots presentes en el escenario, la tendencia lineal observada en los resultados de los estadísticos estudiados, implica que el método planteado es capaz de manejar el incremento de información e interacciones sin desestabilizarse. La tendencia lineal implica que el reparto de la complejidad del escenario entre sus miembros es viable pues, ocupándose cada miembro únicamente de sus interacciones, el resultado del conjunto sigue siendo coherente.

En todos los resultados obtenidos se puede observar que el factor de mayor influencia es el número de robots en la formación, mientras que la influencia de la estructura de formación concreta de cada conjunto de datos queda relegada a un segundo plano.

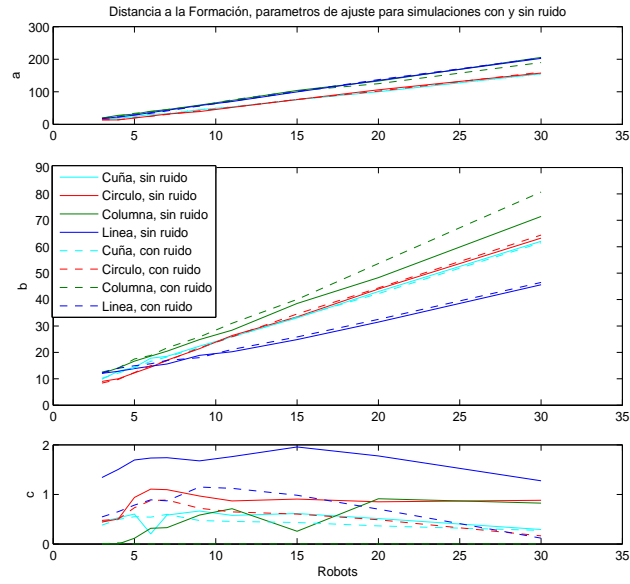


Figura 5.2: Coeficientes de ajuste a un decaimiento exponencial, $d = a \cdot e^{-t/b} + c$, para la evolución de la distancia media de los robots a la formación. Simulaciones con y sin ruido

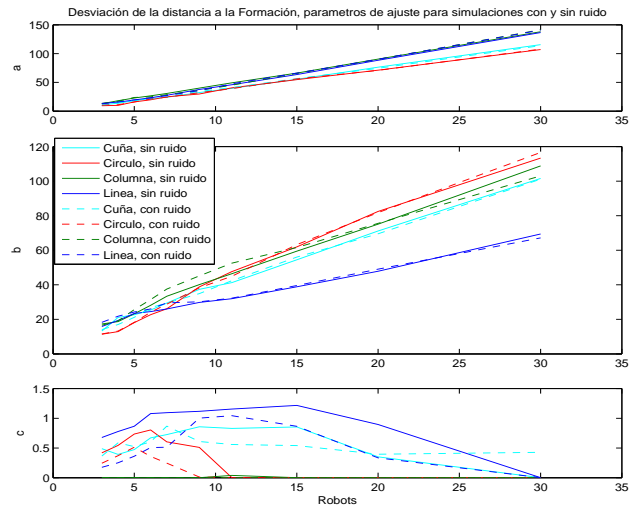


Figura 5.3: Coeficientes de ajuste a un decaimiento exponencial, $d = a \cdot e^{-t/b} + c$, para la evolución de la desviación estándar de la distancia media de los robots a la formación. Simulaciones con y sin ruido

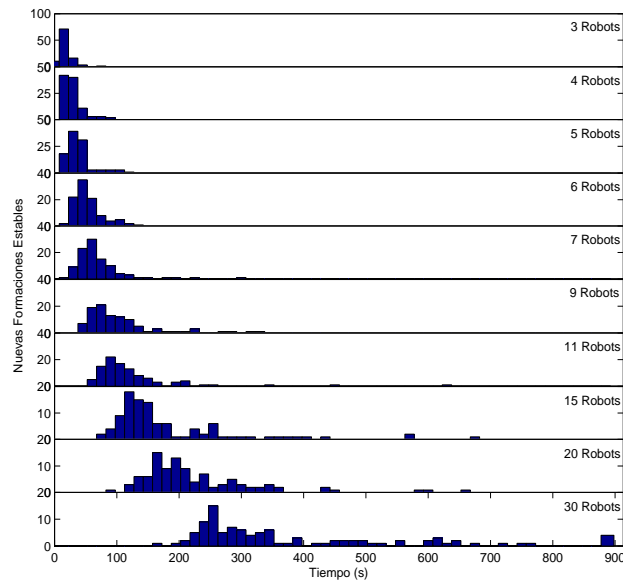


Figura 5.4: Histogramas de tiempo de estabilización de la formación para la formación en cuña, 101 simulaciones en cada caso

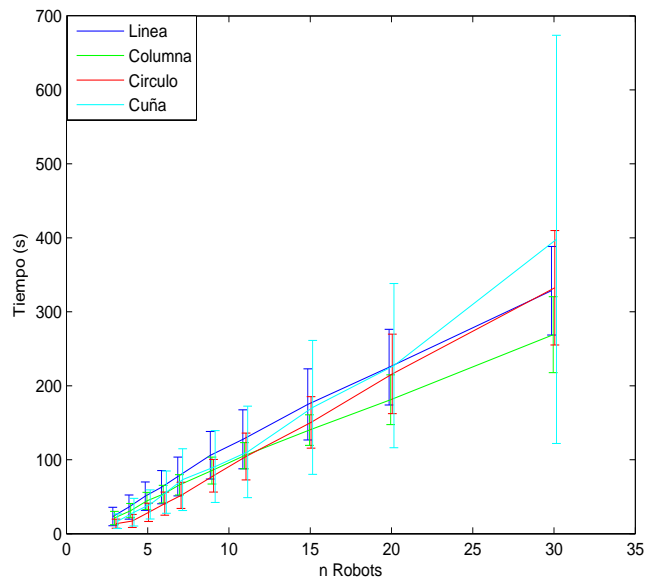


Figura 5.5: Valor medio y desviación para el tiempo de estabilización de la formación

El que la relevancia de la estructura de formación sea secundaria, hace esperar que los resultados sean similares en el caso de grupos de robots con estructuras distintas a las estudiadas y por tanto la validez general del método.

5.2. Robots Reales

Para la prueba del método en robots reales se han llevado a cabo ensayos con diferente número de robots y estructuras de formación. En el estudio sistemático se han realizado un total de 90 ensayos con diferentes estructuras de formación pero siempre con 4 robots.

Los resultados observados en los robots reales, figura 5.6 siguen el mismo patrón que el observado en simulación. Así, realizando los mismo ajustes se observa que los resultados de los robots reales encajan en los resultados obtenidos por simulación, figura 5.7. La principal fuente de discrepancia en el estudio de los robots reales ha sido el error sistemático pero aleatorio introducido por el deslizamiento de las ruedas en el suelo de la zona experimental.

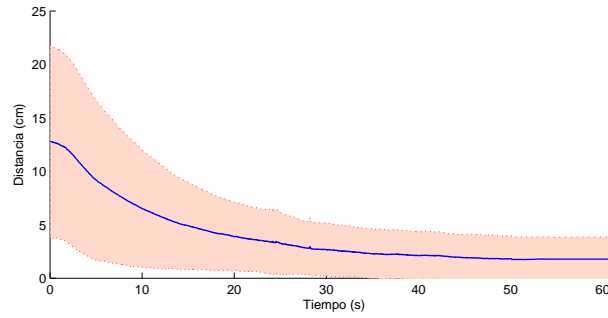


Figura 5.6: Media y desviación de la distancia entre robot y formación a lo largo del tiempo para 4 robots reales en formación de cuña.

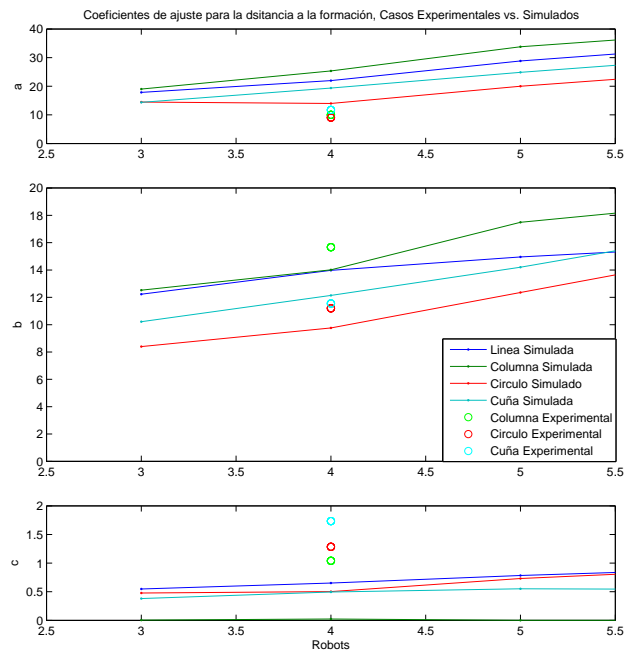


Figura 5.7: Coeficientes de ajuste al decaimiento exponencial, $d = a \cdot e^{-t/b} + c$, de la distancia media entre robot y formación a lo largo del tiempo para 4 robots reales junto a los valores obtenido en simulación.

Capítulo 6

Conclusiones

A lo largo del trabajo presentado se ha establecido que las instrucciones necesarias para guiar al robot en su movimiento se pueden expresar en una forma – más rica que simplemente estableciendo la dirección de movimiento– que permite resolver los conflictos entre las distintas instrucciones.

Mientras que las acciones empleadas a lo largo de este trabajo estaban orientadas a guiar la navegación del robot, el mismo principio de aumentar las dimensiones del espacio de actuación, puede aplicarse a otros ámbitos de la robótica. El uso de un conjunto de acciones mayor que el conjunto de actuaciones disponibles es una forma válida para para expresar las características necesarias de la aplicación y aislar las distintas consideraciones de la aplicación reduciendo las actuaciones conflictivas.

Junto con la forma en la que expresar las acciones, se pone en práctica una forma para combinarlas en su aplicación sobre la actuación final del robot. El método de combinación propuesto está orientado a su uso en robot no holónomos pues son los usados en las simulaciones y las pruebas reales. De todas formas, las diferentes consideraciones tenidas en cuenta para la construcción del método de combinación han sido tratadas de forma independiente, de modo que se puedan extraer y adaptar fácilmente para su uso en otro tipo de robots.

El conjunto de casos estudiados muestra cómo es posible la aplicación sistemática del método propuesto, y cómo es posible mantener aisladas las diferentes consideraciones propias de cada caso de modo que al aumentar el grado del problema estudiado, no sea necesario rehacer completamente el conjunto lógico del robot, si no que es suficiente con añadir los nuevos elementos necesarios en el nuevo problema.

La tendencia lineal con el número de robots en el escenario como factor principal en los estadísticos estudiados –la evolución de la distancia del robot a la formación, la dispersión de los robots en la formación y el tiempo necesario para alcanzar una formación estable– indican que el método propuesto es capaz de manejar la creciente complejidad del sistema de forma sencilla. Así mismo esta tendencia indica que el método propuesto podría emplearse con distinto número

de robots, distintas condiciones iniciales y distintas estructuras relacionales entre robots, de aquellos casos estudiados.

Los resultados de las pruebas hechas mediante robots reales son coherentes con los resultados obtenidos mediante simulación, verificando así la viabilidad del método en su aplicación a casos reales y validando el conjunto de resultados y conclusiones extraídos de la pruebas hechas en simulación.

6.1. Aplicaciones y trabajo futuro

La aplicación más inmediata del método propuesto tiene lugar en el despliegue de equipos de robots autónomos y en tareas donde es necesario mantener una cierta estructura entre los robots.

Sin alejarse del tema de la navegación, la mayoría de las aplicaciones que usan equipos de robots, con un nivel u otro de estructura, podrían beneficiarse del uso del método presentado para la descripción y aplicación de las diferentes tareas a cubrir, por ejemplo, en robots futbolistas.

Si bien el estudio presentado hace uso de robots homogéneos, la metodología propuesta no impone ninguna restricción sobre el uso de grupos de robots heterogéneos. Para ésto sería necesario –de cara al comportamiento de grupo– aislar los factores que son relativos a la navegación del grupo en sí y los factores relativos a la navegación de cada individuo del grupo.

La idea de la ampliación del espacio de acción sobre el espacio de actuación del robot es fácilmente extrapolable a otros dominios de la robótica y de la toma de decisiones en general.

De cara a seguir la actual línea de desarrollo, la navegación en grupo, sería interesante el aumentar la generalización empleada para el modelado del mundo, combinando la descripción actual del mundo –contenida en los campos virtuales de potencial– con diagramas de otro tipo, para conseguir un mayor detalle en la construcción de los mapas de acción o los mapas de decisión –como el uso de histogramas de vectores de campo en la construcción del mapa de regiones de seguridad o del mapa de regiones prohibidas.–

Bibliografía

- [Cifuentes06] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Modular software environment for general purpose autonomous agents simulation and control* in the 5th Mathmod, International Conference on Mathematical Modelling, Viena, Austria, 2006.
- [Cifuentes08a] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Virtual Field Simulation Environment for Robot Groups Navigation Analysis* in the 7th IAS-TED International Conference, Innsbruck, Austria, 2008
- [Cifuentes08b] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Multiple Potentials Approach for Local Navigation of Mobile Robot Groups* in the International Symposium on Robotics, Seoul, Korea, 2008
- [Cifuentes10] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Robot Formation Control Based on a Multi-potential Approach* in the International Conference on Control, Automation and Systems, Gyeonggi-do, Korea, 2010
- [Cifuentes12a] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Robot Navigation Based on Discrimination of Artificial Fields: Application to Single Robots* in Advanced Robotics, Vol. 26 No. 6-7, April 2012.
- [Cifuentes12b] S. Cifuentes, J.M. Giron-Sierra, J. Jimenez *Robot Navigation Based on Discrimination of Artificial Fields: Application to Robot Formations* in Advanced Robotics, Vol. 26 No. 6-7, April 2012.